EXHIBIT 6

UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC., No. C10-4234 SI

Plaintiff,

vs.

JUNIPER NETWORKS, INC.,

/

Defendant.

HIGHLY CONFIDENTIAL - ATTORNEYS' EYES ONLY
DEPOSITION OF: TODD REGONINI
TAKEN ON: AUGUST 14, 2012

REPORTED BY: JODY GIBNEY, CSR NO. 12308, RPR

country, do you know? 13:36:15
A. Yes, I believe all of them are. 13:36:18
Q. Okay. And so Redacted 13:36:22
Redacted in the last two 13:36:26
years, say, 2010 and 2011? 13:36:29
A. I probably can't give an exhaustive 13:36:35
list, but the largest application, you know, what 13:36:40
you could call "marquee applications," would be 13:36:43
T-series platforms, MX platforms, SRX, 13:36:52
potentially some M-series platforms, but that I'm 13:36:56
not sure. 13:36:56
Q. How about J-series routers? 13:36:59
A. I do not know. 13:37:00
Q. One way or the other? 13:37:01
A. One way or the other. 13:37:03
Q. Redacted 13:37:04
Redacted 13:37:07
the T-series, et cetera, in a flow-based 13:37:10
mode. Redacted 13:37:12
MR. KAGAN: Objection; misstates -- 13:37:14
compound. 13:37:14
MR. HOSIE: Q. Do I misstate your 13:37:16
testimony, sir? 13:37:16
Redacted 13:37:19
Redacted

Q. Ahh, thank you. Okay. I'm glad I 13:37:22
asked the question then. My mistake, and I 13:37:24
apologize. It wasn't some clumsy attempt to trap 13:37:28
you. 13:37:28

A. Sure. 13:37:28

Q. Redacted 13:37:31
Redacted you confirmed that 13:37:35
with your call to Jack Redacted 13:37:38

MR. KAGAN: I withdraw my objection. 13:37:40

MR. HOSIE: Thank you. 13:37:40

THE WITNESS: Yes. 13:37:41

MR. KAGAN: Why don't we re-ask. You 13:37:43
want to re-ask it? 13:37:43

MR. HOSIE: Yeah, I will. I apologize. 13:37:46

Q. So you called Jack Redacted 13:37:52
Redacted 13:37:54
Redacted and you said, To your 13:37:58
knowledge, Jack, Redacted 13:38:01
Redacted in flow-base, and he 13:38:04
confirmed that they were not. Fair summary? 13:38:07

MR. KAGAN: I think it misstates the 13:38:09
testimony. 13:38:09

THE WITNESS: So I called Jack and asked 13:38:12
him specifically about the marquee applications 13:38:15
Redacted

MR. HOSIE: Q. Yes. 13:38:17

A. And specifically whether they 13:38:19
included that MSPTC hardware, M-S-P-T-C hardware 13:38:27
and/or used security-based services or not, and 13:38:30
he confirmed that they do not. 13:38:31

Q. Redacted 13:38:33
Redacted
13:38:37

A. They have. 13:38:37

Q. And do they use them for flow-based 13:38:40
security? 13:38:40

A. There is -- there are a small number 13:38:48
that I'm aware of that do. 13:38:49

Q. And by "small number," what do you 13:38:51
mean? 13:38:51

A. I mean in numbers around 50 systems 13:39:02
versus the primary platform in that network being 13:39:09
on the order of a couple thousand. So in the 13:39:11
network application that I'm -- that I'm thinking 13:39:14
of, this handful of boxes, as I would call it, 13:39:18
the 50 versus a much larger number, would be the 13:39:21
ones that leveraged the -- it's actually for 13:39:23
network address translation. 13:39:25

Q. Ahh, so that has to be session-based 13:39:27
for NAT?

A. In our implementation it is, yes. 13:39:30
Q. Redacted 13:39:34
Redacted in a flow-based way for 13:39:37
network address translation? 13:39:39
A. In that application, yes. 13:39:40
Q. Redacted 13:39:42
Redacted 13:39:44
A. I can't say. 13:39:44
Q. Is it tens of millions, hundreds of 13:39:48
millions? 13:39:49
A. I can't say, honestly. 13:39:50
Q. Are they located in this country? 13:39:53
A. I believe all of them are. 13:39:55
Q. Okay. And of the -- you said there 13:40:03
were approximately 2,000 boxes in the application 13:40:05
of which only 50 are yours. Who -- 13:40:09
MR. KAGAN: Misstates testimony. 13:40:10
MR. HOSIE: Q. Is that not true? 13:40:12
A. No, I didn't say 50 were ours, 13:40:16
actually. The specific application I was 13:40:18
thinking of was the Redacted 13:40:22
Redacted 13:40:22
Q. Okay. 13:40:23
A. -- which is also all Juniper 13:40:24
equipped.

I, Jody Gibney, R.P.R., C.S.R. No. 12308, a Certified Shorthand Reporter in and for the County of Marin, State of California, do hereby certify:

That the witness named in the foregoing deposition, Todd Regonini, was duly sworn by me.

That said deposition was taken before me at the time and place set forth and was taken down by me in shorthand and thereafter reduced to computerized transcription under my direction and supervision, and I hereby certify the foregoing deposition is a full, true and correct transcript of my shorthand notes so taken.

I further certify that I am neither counsel for nor related to any party to said action nor in any way interested in the outcome thereof.

IN WITNESS WHEREOF, I have hereunto subscribed my name this ___ day of ________________, 2012.

____

Jody Gibney
Certified Shorthand Reporter No. 12308, RPR

EXHIBIT 7

Page 1

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC.,	)	
	)	
Plaintiff,	)	
	)	
vs.	)	No. C10-4234 SI
	)	
JUNIPER NETWORKS, INC.,	)	
	)	
Defendant.	)	
	)	

CONTAINS CONFIDENTIAL PORTION

VIDEOTAPED DEPOSITION OF SCOTT NETTLES, PH.D.
San Francisco, California
Friday, October 19, 2012
Volume I

Reported by:
SUZANNE F. BOSCHETTI
CSR No. 5111

Job No. 1540467

CONFIDENTIAL PORTION: 165 - 199
PAGES: 1 - 297

Page 121

representation -- I mean I don't remember if this document specifically is -- is cited in Dr. Calvert's report and -- and in my report. This might be under the title sort of Cisco documents, but I don't remember disputing that this was prior 01:37:02 art.

Q And if you turn to the second page of the exhibit, there's a Figure 1. And then a sentence that reads:

"With NetFlow Switching, this 01:37:20 process occurs only with the first packet in a flow."

Do you see that there?

A Yes, sir, you read that correctly.

Q Do you have any reason to believe that that 01:37:28 is not an accurate statement of Cisco technology that was available at the time of this paper?

MR. HOSIE: Objection. Overbroad. Vague and ambiguous.

THE WITNESS: I mean, I haven't read this 01:37:39 document in -- in detail now. And if I read it in detail before, it was a long time ago, and I've read lots of documents. So, you know, it's hard to agree or disagree with one single statement taken out of one Web page. And I mean the Web page looks like 01:37:58

Page 122

it's sort of marketing material, so I'm always dubious about marketing material.

BY MR. MCPHIE:

Q Why are you dubious about marketing material? Well, let me withdraw that. 01:38:12

In -- in what sense are you dubious about marketing material in coming to your opinions in this case?

MR. HOSIE: Objection. Vague and ambiguous, "marketing material." 01:38:23

THE WITNESS: Well, it's just that marketing material is generally high level. It's usually meant for a lay audience, so it's often not very precise. I know that when I've been an engineer for a company, that sometimes the way the 01:38:40 company's marketing arm describes the things that I know a lot about, you know, it's not -- I mean I'm not saying that it's -- it's a lie, but it's not necessarily the most technically accurate. So, you know, I sometimes take that with a grain of salt. 01:39:01

Sometimes if a company represents something to be true a lot and in a forceful way, I take that more seriously. Here I don't think they're -- they're not telling all their customers this. Only a few customers care about this question, so -- 01:39:19

Page 123

You know, if you want to show me the Kerr patent and show how this corresponds to the Kerr patent, then I would certainly take it that the representations they make to the patent office are truthful. 01:39:31

BY MR. MCPHIE:

Q But in general, a marketing piece is going to be less reliable than, say, a more detailed technical document, fair?

MR. HOSIE: Objection. Overbroad. Vague 01:39:49
and ambiguous.

THE WITNESS: No, I didn't -- I didn't say that. I -- I said that you have to potentially give marketing material more scrutiny, and I said that the extent to which a representation that is 01:40:03
marketing-oriented is made might influence that. Also, the extent to which a marketing-oriented statement is made in technical documents might also influence that, you know. So, you know, I think we have an example of that in this case. 01:40:21

But anyway, the main point about this sentence is just that I haven't read this carefully, and so, you know, I -- I haven't -- I don't know the details of this system, so I'm just not going to necessarily agree that this sentence is true 01:40:36

Page 164

VIDEO OPERATOR: We are back on the record at 2:47 p.m. This marks the beginning of volume 1, media No. 3 of the deposition of the Dr. Scott Nettles. Please continue.

MR. MCPHIE: I would like to take the 02:47:18
opportunity now to, because of the next line of questions we're going to go into, to mark this portion of the transcript under the highest level of confidentiality under the protective order pertaining to source code. 02:47:33

(Continuing testimony is contained in the Highly Confidential - Source Code section, pages 165 - 199.)

---oOo---

Page 165

---oOo---

(Deposition Exhibit 222 marked by the court reporter.)

(Deposition Exhibit 223 marked by the court reporter.) 02:48:09

BY MR. MCPHIE:

Q Dr. Nettles, you now have in front of you two exhibits. One is entitled "JUNOS Analysis" and is Exhibit 222. It bears the Bates No. IMP141451. And the other is a collection of documents. This is 02:48:39 Exhibit 223, a collection of documents beginning with the Bates No. IMP141496.

Please take a look at these documents with a sufficient level of scrutiny to tell me whether you've seen them before. 02:49:07

A These appear to be the documents that I got from Pavel concerning his analyses. Obviously mine weren't Bates stamped, so I can't know absolutely certain, but I'm assuming given the Bates number that you got them from Implicit and that they're 02:50:44 those documents.

Q And when you refer to Pavel, you're referring to Pavel Treskunov, correct?

A I hope so.

Q And I believe we referred to him in the 02:50:55

Page 166

last deposition as simply Pavel?

A That -- that was my intention by calling him Pavel in this one.

Q Okay.

A I apologize. I'm being bad with names. 02:51:04

Q No trouble at all. Initially what I'll ask -- well, withdrawn. Were there any other documents that Pavel Treskunov authored that you received or considered in connection with the preparation of your expert report in this case? 02:51:29

A I think that probably some parts of the most recently served infringement contentions were written by Pavel. I'm not positive about that. But to the extent that any of those were, I certainly received those and -- and utilized those in 02:51:56 preparing my report. But I don't remember any other documents beyond that.

MR. HOSIE: Counsel, we can represent that all of the Pavel documents provided to Dr. Nettles have been produced in the litigation. 02:52:08

BY MR. MCPHIE:

Q And therefore other than Exhibit 222 and 223, there are no other documents from Pavel Treskunov that were provided to Dr. Nettles?

MR. HOSIE: I'd have to go back and verify 02:52:23

Page 168

say marginal grammar. But that's not to say that I'm representing that I agree with everything in -- in these documents either. It's just been a long time since I reviewed them. But I -- I don't remember reading it and thinking oh, that must be 02:54:23 wrong.

BY MR. MCPHIE:

Q Did you receive all of these documents, Exhibit 222 and 223, all at once, or would pieces come to you over the course of the work you did on 02:54:36 this case?

A I think I received them all at once but, you know, as you know, there's a lot of documents in a case like this, so it's a little hard to remember exactly when I -- when I received what when. But I 02:54:57 think probably all at once.

Q Can you say approximately when you would have received Exhibits 222 and 223?

A It would either have been when I -- when I signed the protective order and I was cleared, then 02:55:44 I got a bunch of confidential documents involving this case. And I don't remember exactly when that was. Your records would probably reflect that more closely than -- than my remembrance, because you would have had the final say in clearing me. Either 02:56:06

Page 169

then or it would have been sort of mid or maybe two-thirds of the way through July. And I just -- I don't -- I don't remember. I remember getting a bunch of documents once I was -- once I was cleared under the protective order. 02:56:28

Q Did Pavel Treskunov ever communicate to you that all of the Juniper accused products operated essentially in the same manner as the multi-services module for purposes of infringement?

MR. HOSIE: Objection. Vague. Ambiguous. 02:56:53 Overbroad.

THE WITNESS: Pavel never communicated that to me. My understanding that that was the case came from reading Juniper documentations and looking at Juniper diagrams and Juniper's representations to 02:57:10 this effect. And maybe it says it in here, but I don't remember it saying it in here.

BY MR. MCPHIE:

Q You understood that Exhibit 222 was an analysis that was primarily focused on the 02:57:37 multi-services module, correct?

MR. HOSIE: Objection. Vague and ambiguous, mischaracterizes the exhibit.

THE WITNESS: Well, I mean, again, based on Juniper's representation to the public and the many 02:58:00

Page 170

diagrams that were -- are very similar in Juniper publications across all of these different product lines, my understanding was in general that this was broadly applicable. I mean obviously, for example, there's a -- there's a specific reference here that 02:58:26
I see to Multi-services daemon.

BY MR. MCPHIE:

Q Which page?

A Oh, sorry. Page 8.

Q What is the significance of the 02:58:46
Multi-services daemon reference on page 8?

MR. HOSIE: Objection. Lacks foundation.

THE WITNESS: Well, this looks like this is talking about sort of the -- the main loop that drives the whole -- the whole thing. And because 02:58:59
now I understand that this use of this word multi-services often refers to these MultiServices PICs, I think I see that this is probably a reference to that particular system.

But, you know, all of this information 02:59:22
comports with my understanding and Juniper's representations of the way the systems operate for the accused products. So I don't think that -- you know, to me it seems like much of this material is completely applicable to all of the accused 02:59:40

Page 174

the report.

THE WITNESS: Yes, I know.

MR. HOSIE: That's where all the technical analysis is.

THE WITNESS: No, I understand. It's not 03:04:21
the right part of this report.

MR. HOSIE: There you go.

MR. MCPHIE: Yes, I -- I hear what you're saying.

This I believe was Exhibit 206. 03:04:32

(Previously marked Exhibit 206 was presented to the witness.)

BY MR. MCPHIE:

Q So if you look at page 14 of Exhibit 206,
the top of the page says "JNI's basic packet 03:05:20
processing loop." Do you see that there?

A Yes, sir, I do.

Q And JNI in this instance, you're referring to Juniper Networks Inc.?

A That's correct. 03:05:34

Q And then below that you state, "Put graphically, JNI's basic packet processing loop is as follows."

A Yes, sir.

Q And then there's a graphic which is 03:05:46

Page 175

essentially identical to the graphic we were looking at on page 8 of Exhibit 222. This is the Pavel Treskunov report?

A Yes, sir, I see that.

Q And you see that the top part of page 8 of 03:06:02
the Pavel Treskunov report in fact is not included in your expert report, correct?

A Well, I see it's not included right there. I'll take your representation that it's not included in general. 03:06:26

Q So again, my question is, why did you copy the main packet loop diagram but not the heading regarding Multi-services daemon?

MR. HOSIE: Objection. Mischaracterizes the document. 03:06:44

THE WITNESS: Well, I don't remember precisely why I didn't copy this particular paragraph. But when I read the paragraph, I would have been surprised that I would have, because all this paragraph is describing is how this daemon 03:07:13
basically spawns this loop onto every CPU that's part of the systems. These are typical multi-CPU systems.

And so the fact that this loop runs on many CPUs isn't really particularly important to the 03:07:28

Page 176

infringement analysis. In fact, it's not important to the infringement analysis at all that I can recollect. And so that -- that paragraph just doesn't contribute anything to the analysis. I -- I don't remember if that was my -- my thinking or if 03:07:48 my thinking was really sort of the opposite, well, what's the main stuff.

BY MR. MCPHIE:

Q Was it intentional that you left off the paragraph at the top of page 8 from your 03:08:01 infringement report?

MR. HOSIE: Objection.

THE WITNESS: I mean, I think I just answered that, and I answered that I don't remember why exactly I did it. But if I read that paragraph 03:08:19 now, I don't think that that paragraph actually contributes to my infringement analysis, so it might have been something that I actually made a decision about. So it would have been intentional. But there's nothing -- there's nothing informative about 03:08:38 that paragraph. All it really says is we're going to run this loop on every CPU. That's all.

BY MR. MCPHIE:

Q And, in fact, the paragraph at the top of page 8 of Exhibit 222 explains that the main packet 03:08:52

Page 177

loop illustrated below is part of the Multi-services daemon, correct?

A It does. What's not clear is that the Multi-services daemon is necessarily tied to the MultiServices PIC. 03:09:15

Q Turn a couple of pages earlier in Exhibit 222.

A Which page?

Q Page 6.

A Okay. 03:09:31

Q Do you have that there?

A I do.

Q And there's an illustration, Figure 2 in the middle of the page. Do you see that as well?

A Mm-hmm. 03:09:43

Q Now, this figure is not reproduced in your expert report, correct?

A I'd have to look, but not that I remember.

Q And the label to the figure is:

"Figure 2: Packet Flow through the 03:10:04
Adaptive Services or MultiServices PIC."

Do you see that?

A I do.

Q And then in the figure itself there's a box labeled "Adaptive Services or MultiServices PIC." 03:10:20

Page 179

(Reporter's clarification.)

Sorry, I apologize.

"JUNOS Software is a single network
operating system integrating routing,
switching, and security. Most Juniper 03:12:18
Networks hardware platforms run JUNOS
Software (herein JUNOS)."

Then it goes on to talk a little bit more
about JUNOS. And we know for a fact that in
addition to the MultiServices PICs -- I mean, 03:12:33
they're part of a router -- in addition to those
routers running JUNOS, that the J series routers and
the SRX series routers run JUNOS. So, you know,
that seems clear that Pavel -- the understanding
here was that this was about how JUNOS worked. 03:12:50

Q And, in fact, it was clear in your mind
upon carefully reviewing Exhibit 222 that the
analysis, the detailed analysis of Exhibit 222
applied to each and every one of the Juniper accused
products, right? 03:13:17

MR. HOSIE: If I could have that read back,
please.

MR. MCPHIE: I can read it.

BY MR. MCPHIE:

Q And, in fact, it was clear in your mind 03:13:32

Page 180

upon carefully reviewing Exhibit 222 that the detailed analysis in the exhibit applied to each and every one of the accused Juniper products, correct?

MR. HOSIE: Objection. Vague. Ambiguous. Overbroad. 03:13:56

THE WITNESS: You know, I don't remember exactly what was clear in my mind. This is many months ago. I've done a lot of work on this case and other technical things. But it's clear to me that Juniper represents that there's one JUNOS, and 03:14:13 it's clear to me that the -- the general description of what's going on here matches and is completely consistent with the general description of what happens in Juniper's products, the J series and the SRX series. 03:14:39

For example, this -- this data path here, you know, it matches up with that canonical picture that Juniper publishes for essentially any of its products that involve flow-based processing. So I think it's very consistent with what Juniper says in 03:14:58 Juniper's documentation. But I don't remember exactly what was clear to me in my mind at the time.

BY MR. MCPHIE:

Q Is it clear to you now upon reviewing Exhibit 222 that in fact the detailed analysis of 03:15:14

Page 181

Exhibit 222 applies to each and every one of the Juniper accused products?

MR. HOSIE: Objection. Asked and answered. Vague and ambiguous. Overbroad.

THE WITNESS: Well, what's clear to me now 03:15:48
is that the functionality that's described in this document is reproduced in the -- particularly the accused products. Again, some of this functionality doesn't apply to dumb routers that don't support services. 03:16:08

Now, some of your experts report -- calls into question whether or not the specific code sites come from the code base for the J series or SRX. I haven't had a chance to verify that myself. So I certainly am -- you know, have some reason to be 03:16:27
less certain about that.

But I certainly have no reason to not be certain that the functionality is the same. And that's what's really important for the infringement analysis is what the functionality does. And this 03:16:43
comports with everything I know about the functionality of the J series and SRX series.

BY MR. MCPHIE:

Q And you feel confident testifying today --

MR. HOSIE: Could you speak up, David? 03:16:54

Page 182

BY MR. MCPHIE:

Q And you feel confident testifying today under oath that in fact you understand the detailed analysis of Exhibit 222 to apply equally to each and every one of the accused Juniper products in this 03:17:13 case, correct?

MR. HOSIE: Same objections plus just asked and answered.

THE WITNESS: I mean that just mischaracterizes what I said. What I said is that 03:17:26 because of things that were in your expert's report, I'm less certain that all of the code modules that are cited in this report, and I think their code modules may be cited in this report as well, that they're part of the code base for SRX and the J 03:17:47 series routers. And that's despite Juniper's representations to the contrary to its clients.

But -- so I'm not as confident about that, so I'm certainly not going to testify that I'm absolutely sure that every one of these pieces of 03:18:04 code that are cited applies to the SRX or the J series. But the basic overall functioning is completely consistent with everything in my report about the J series and the SRX series. So I certainly don't think that, you know, there's -- I 03:18:22

Page 183

don't see a contradiction there, so --

BY MR. MCPHIE:

Q Turn to page 4 of Exhibit 222.

A Okay.

Q And you see there's a description along 03:18:43
with a graphic regarding a fundamental division with a control plane, a data plane, and a service plane. Do you see that there?

A Yes, sir.

Q Turn to the next page. The third complete 03:19:03
paragraph contains a statement beginning at the end of line 2:

> "The services plane runs on optionally installable and hot swappable hardware, which are generically called 03:19:25
> MultiServices (MS) modules."

Do you see that sentence there?

A Yes, sir, I do.

Q Is that an accurate statement --

MR. HOSIE: Objection. 03:19:39

BY MR. MCPHIE:

Q -- in your opinion?

MR. HOSIE: Objection. Overbroad. Vague and ambiguous.

THE WITNESS: Well, I don't know exactly 03:19:50

Page 184

what -- what document is being quoted here. I'd have to look back and -- and see, because this is obviously some information -- he's quoting from some Juniper document. But my understanding is that in the routers that support the MultiServices PICs, 03:20:05 and -- there's two kinds of PICs. I'm trying to remember what they're both named, but they're -- they both support this MultiServices. But this is an accurate statement.

My further understanding is that this same 03:20:20 functionality is found in the J series routers, but it's not optional. Although whether or not you use it to do services or just to be packet processing, that's an option.

And further that this same functionality 03:20:36 appears in the low end three digit SRX boxes. Again, it's not optional. That's how those boxes are designed. It's built into those boxes. And again, there's a choice in those boxes between packet processing and flow-based processing. 03:20:54

And then in the high end SRX boxes, the four digit ones, again the same functionality is there, but again it's not optional but the same functionality is there.

So this statement is really talking about 03:21:09

Page 185

the PICs. That's where there's optional hardware that you can take in and out.

BY MR. MCPHIE:

Q Is it fair to say that this -- well, withdrawn. 03:21:24

Is it fair to say that you would agree that this paragraph regarding the services plane on page 5 of Exhibit 222 is not talking about the SRX or J series products?

MR. HOSIE: Objection. Vague and 03:21:42 ambiguous.

THE WITNESS: No, sir, I -- I think that's -- I think that's not correct. My understanding is that this model that we saw on the previous page where there's a control plane, a data 03:21:55 plane, and a services plane, applies to the J and SRX. It's just that this one sentence that we were focusing on a second ago about it running on optional hardware, so in those devices it's just not optional. But there's certainly a notion that 03:22:13 there's a data plane and a services plane. That's a -- that's a general abstraction. It takes -- takes a cue from standard networking thinking where there's a control plane and a data plane to add the idea that there's a services plane. And I think 03:22:35

Page 186

that model is actually very applicable to all the products that support services at all.

BY MR. MCPHIE:

Q Does the services plane on the SRX product run on an optionally installable and hot swappable 03:22:48 hardware generically called MultiServices modules?

MR. HOSIE: Objection. Asked and answered.

THE WITNESS: I already answered this question, so I explained to you that that particular sentence is applicable to the MultiServices PIC. 03:23:05 And again, there's something called an AS PIC, which -- I think it's an AS PIC. Anyway, there's several -- there's several PICs that run services where the hardware is optional, but the devices that were ultimately accused are the ones where it's not 03:23:22 optional.

BY MR. MCPHIE:

Q So you would agree this one sentence that talks about optionally installable and hot swappable hardware is not talking about the SRX or J series 03:23:34 products, correct?

A There is optionally installable hardware on the SRX series, especially on the high end ones, but they're not -- that's not the hardware they're talking about here. 03:23:53

Page 187

Q The MultiServices module is not used on the SRX and J series products, correct?

A Yes, sir, that's correct.

Q And you knew that at the time that you prepared your expert report regarding infringement 03:24:08 in this case, correct?

A Yes, sir, I don't think there's any representation in my report that it is used in those products.

Q In the next paragraph there's another 03:24:20 reference to "Each MS module." Do you see that there?

A Yes, sir.

Q And then go a couple of paragraphs down, there's a statement: 03:24:31

"First the service software must be installed on MS modules."

Do you see that there?

A Yes, sir.

Q And in fact, that same paragraph has two 03:24:43 more references to an MS module, correct?

A Yes, sir.

Q And by the way, you understand that MS module to be an abbreviation for the MultiServices module, correct? 03:24:58

Page 188

A I do. I assume that's a generic term for these two different MultiServices PICs.

Q And indeed the MultiServices module is another name for the MultiServices PIC, correct?

A Well, again, I think there's -- the best of 03:25:16 my recollection, and this isn't in my report because they weren't eventually accused, but there are two different MultiServices PICs, one of which I think is called the MultiService PIC, and I think that here whoever authored this is using this MS module 03:25:31 as a generic way of not having to talk about both of them, to talk about them together.

Q So you understand the term MS module to include both the MultiServices PIC as well as the Adaptive Services PIC, correct? 03:25:48

A That's it. Adaptive Services. I knew it was AS, but -- yes, sir, that's my -- again, I'd have to read this document more carefully, but that's my best guess.

Q Going on to the next paragraph, there's 03:25:59 another sentence that talks about "once the service software starts on an MS module." Do you see that there?

A Where? Oh, okay, I see it.

Q You have that? 03:26:13

Page 189

A Mm-hmm.

Q Okay. And then finally, the next paragraph there's also a reference to the MS module, correct?

A Yes, sir.

Q And if you turn to page 6, there are also a 03:26:24
number of references to MS modules in the bullet points at the top of the page. Do you see that?

A I do.

Q In the second bullet point, second sentence, there's a statement: 03:26:43

> "A service set captures one or more services' policies to be applied and an MS module to which to redirect packets for servicing."

Do you see that? 03:26:57

A I see something about steering. Oh, yes, I see that.

Q So this language is now connecting the concept of the service set to an MS module, correct?

A In this particular description, which we 03:27:19
don't really know where it came from, but yes.

Q And if you look at the diagram on paragraph 6, there's a box, I think we looked at it earlier, that says "Adaptive Services or MultiServices PIC." Do you see that? 03:27:38

Page 190

A Yes, sir, I do.

Q And then there's an arrow that comes up and there's a 2 by the arrow. Do you see that?

A I do.

Q And it points to something -- well, the 2 03:27:46 is underneath something called "Interface service set." Do you see that there?

A I do.

Q And then also if you look at No. 4 on the diagram, there's something else called "Interface 03:27:59 service set." Do you see that?

A I do.

Q And that has an arrow pointing directly to "Adaptive Services or MultiServices PIC." Do you see that? 03:28:11

A I do.

Q So again, this diagram is connecting the concept of service sets with the adaptive services or MultiServices PIC, correct?

A In this particular document it seems like 03:28:24 the service sets are somehow handled by the adaptive services or MultiServices PICs, that's correct.

Q Why did you choose not to use Figure 2 on page 6 of Exhibit 222 in your expert report?

A Well, I mean, the best of my recollection 03:28:47

Page 191

was that there was a decision made to -- so I'm not really sure exactly the process, but I'm pretty sure the MultiServices PICs were part of the original infringement contentions, and there was a decision made to drop them as accused products. 03:29:15

And this picture in particular and the discussion that we've been looking at here is clearly directed to those products. And so I -- I would have dropped them because I understood them not to apply necessarily to the J routers and the 03:29:35 SRX routers.

It's not to say that all of these concepts don't appear in those products, but, you know, this figure in particular that you ask about, it says "Packet Flow through the Adaptive Services or 03:29:52 MultiServices PIC." If I had put that in the report, you would have been asking me well, why did you put a picture in that applies to the adaptive services and MultiServices PIC because it's not accused. So that's why I didn't put it in. 03:30:04

Q In your opinion is any of the discussion on page 5, 6, 7 and the top of page 8 of Exhibit 222 relevant to understanding the discussion that follows regarding the main packet loop?

MR. HOSIE: Objection. Vague. Ambiguous. 03:30:28

Page 195

literature -- I've reproduced that figure several different times from several different sources -- then that might be a fair inference.

But I think given Juniper's representation about there being one operating system and Juniper's 03:39:03 representation about how that operating system processes flows, that no, I don't think that it's -- it's clear at all that this is limited to the MultiServices PIC.

I don't know exactly where Pavel took this 03:39:21 text. I'm guessing that he cut and pasted this from some Juniper document that he hasn't cited to here because it reads like documentation written by a computer company, not a description written by a person whose first language isn't English. So I'm 03:39:41 guessing that this description is something from a Web page or a book or -- or something.

It might even be from the Enterprise Routing book. There's a lot of discussion of this sort in the Enterprise Routing book. We could go 03:39:56 and look. I think the Enterprise Routing book makes it clear that the J series and SRX series and the services PICs all process services and flows in a similar way.

BY MR. MCPHIE: 03:40:12

Page 196

Q The figure on page 6 of Exhibit 222 is substantially different from the figure that appears on the top of page 10 of Appendix A of your report, correct?

MR. HOSIE: Objection. Vague and ambiguous. 03:40:25

THE WITNESS: Well, it would appear to be at first blush -- topologically it's quite different. But if I look at it, I see actually it's quite -- it's quite similar. So the figure on 03:40:42 page 10 on the left-hand side, it shows a packet, it shows some kind of input to a series of boxes there. I would understand this -- I mean I know the context of this figure to be a packet being put into a queue. And that would correspond to getting the 03:41:07 next incoming packet.

Then there's something called per packet policers and shapers, and that, for example, would correspond to header integrity check and IP fragmentation check. That's a per packet. 03:41:23

BY MR. MCPHIE:

Q What are you looking at?

A I am looking at the -- you told me that the loop -- you told me that this loop --

Q No, no, no. I'm sorry if I wasn't clear. 03:41:36

Page 197

I'm referring to on the top of page 10.

A You said that this loop and this loop on top of page 10 are very different.

Q No, no, no.

A What did you say was very different? 03:41:47

Q What I'm pointing to is page 6 of Exhibit 222 --

MR. HOSIE: Why don't you give us a Bate number by page.

BY MR. MCPHIE: 03:41:56

Q Bates No. IMP141546 and the diagram on the top of page 10.

A But I -- I never said that page 6 -- this diagram on page 6 that's labeled Figure 2 in Pavel's report that we've been talking about is similar to 03:42:11 the picture in my report. I said that --

Q Okay. I'll ask --

A I said that the picture on page 8 is similar.

Q I'll ask the question. The figure on 03:42:19 page 6 of Exhibit 222 is substantially different from the figure on the top of page 10 of Appendix A to your report, correct?

MR. HOSIE: Objection. Vague and ambiguous. 03:42:39

Page 297

I, the undersigned, a Certified Shorthand Reporter of the State of California, do hereby certify:

That the foregoing proceedings were taken before me at the time and place herein set forth; that any witnesses in the foregoing proceedings, prior to testifying, were duly sworn; that a record of the proceedings was made by me using machine shorthand which was thereafter transcribed under my direction; that the foregoing transcript is a true record of the testimony given.

I further, certify I am neither financially interested in the action nor a relative or employee of any attorney or party to this action.

IN WITNESS WHEREOF, I have this date subscribed my name.

Dated: 10/26/12

SUZANNE F. BOSCHETTI
CSR No. 5111

EXHIBIT 8

APPENDIX A
JNI Technology Overview and Elements

Table of Contents

OVERVIEW OF JNI TECHNOLOGY

1. This section overviews JNI's technology to support the claim-by-claim limitation-by-limitation analysis below. There may be some information that is not specifically about Juniper's technology, but that information also supports those arguments. Some important additional technical background is found in the Technology section of the main report.

A. Technical Aspects of the Accused Systems Hardware

2. Certain aspects of the infringement proof below require certain details concerning the hardware of the Accused Products. In particular, that the hardware is designed to receive packets and has computer readable media that is not data transmission media. In addition, I establish that certain aspects of the high-end SRX products are fundamental to the infringement.

3. Junos Security, a text cited in Exhibit 2, has significant discussion of the SRX hardware. This discussion details the general design of Juniper products and makes it clear that these products are designed to send and receive packets. Discussion of the SRX100 line starts at page 30, the SRX200 line at page 32, of the SRX600 line at page 36, a general overview of the branch SRX Hardware (3 digit) at page 42, general discussion of the Data Center SRX (4 digit) devices at 46, and further overview of the Data Center line at 55. In general all of Chapter 1, pages 1-69, provides details about the SRX products, how they are used, and their basic function.

4. Junos Security also makes it clear that considering only the base chassis for the purposes of considering the economic value of the system lacks any technical support as more than just the base chassis is fundamentally a part of infringement.[1] Perhaps the most telling except is from page 46:

[1] I understand that Implicit took a Rule 30(b)(6) per Court Order deposition on August 14, 2012, and I expect to amend this report to include information from that deposition when the transcript is available.

> Data Center SRX Series The data center SRX Series product line is designed to be scalable and fast for data center environments where high performance is required. Unlike the branch products, the data center SRX Series devices are highly modular— a case in point is the base chassis for any of the products, which does not provide any processing power to process traffic because the devices are designed to scale in performance as cards are added. (It also reduces the total amount of investment that is required for an initial deployment.)

And concerning the SPC (page 48):

> The element that provides all of the processing on the SRX Series is called the Services Processing Card (SPC). An SPC contains one or more Services Processing Units (SPUs). The SPU is the processor that handles all of the services on the data center SRX Series firewalls, from firewalling, NAT, and VPN to session setup and anything else the firewall does.

And concerning the NPU (page 49):

> The NPU or Network Processing Unit is similar in concept to the SPU, whereby the NPU resides on either an input/ output card (IOC) or its own Network Processing Card (NPC) based on the SRX platform type (in the SRX5000 line the NPU sits on the IOC and in the SRX3000 line it is on a separate card). When traffic enters an interface card it has to pass through an NPU before it can be sent on for processing. The physical interfaces and NPCs sit on the same interface card, so each interface or interface module has its own NPU. In the SRX3000 line, each interface card is bound to one of the NPUs in the chassis, so when the SRX3000 line appliances boot, each interface is bound to an NPU in a round-robin fashion until each interface has an NPU. It is also possible to manually bind the interfaces to the NPUs through this configuration.

And finally discussion of the roles of the NPU and SPC in session creation, a basic aspect of infringement, is found on pages 46-57.

5. The Dynamic Services Architecture whitepaper (http://www.juniper.net/us/en/local/pdf/whitepapers/2000288-en.pdf) pages 4-7 also has significant discussion of how the SPC and NPU function and the central role they play in the infringement of Data Center SRX products.

6. Datasheets and other documentation provided by JNI also provide evidence both concerning the basic hardware of the Accused Products, that they operate as networking devices

and send and receive packets and that they have computer readable media that is not data transmission media both in the form of flash memory and of "main" computer memory. The following list is just a small subset of the information that Juniper has on its website:

- http://www.juniper.net/us/en/local/pdf/datasheets/1000281-en.pdf
- http://www.juniper.net/us/en/local/pdf/datasheets/1000254-en.pdf
- http://www.juniper.net/us/en/local/pdf/datasheets/1000267-en.pdf
- http://www.juniper.net/us/en/local/pdf/datasheets/1000336-en.pdf
- http://kb.juniper.net/InfoCenter/index?page=content&id=KB24309&cat=security
- http://kb.juniper.net/InfoCenter/index?page=content&id=KB24309&cat=security_products&actp=LIST&showDraft=false
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/pathway-pages/hardware/srx100/index.pdf
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/pathway-pages/hardware/srx110/index.pdf
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx1400/hw/srx1400-hardware-guide.pdf
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/pathway-pages/hardware/srx210/index.pdf

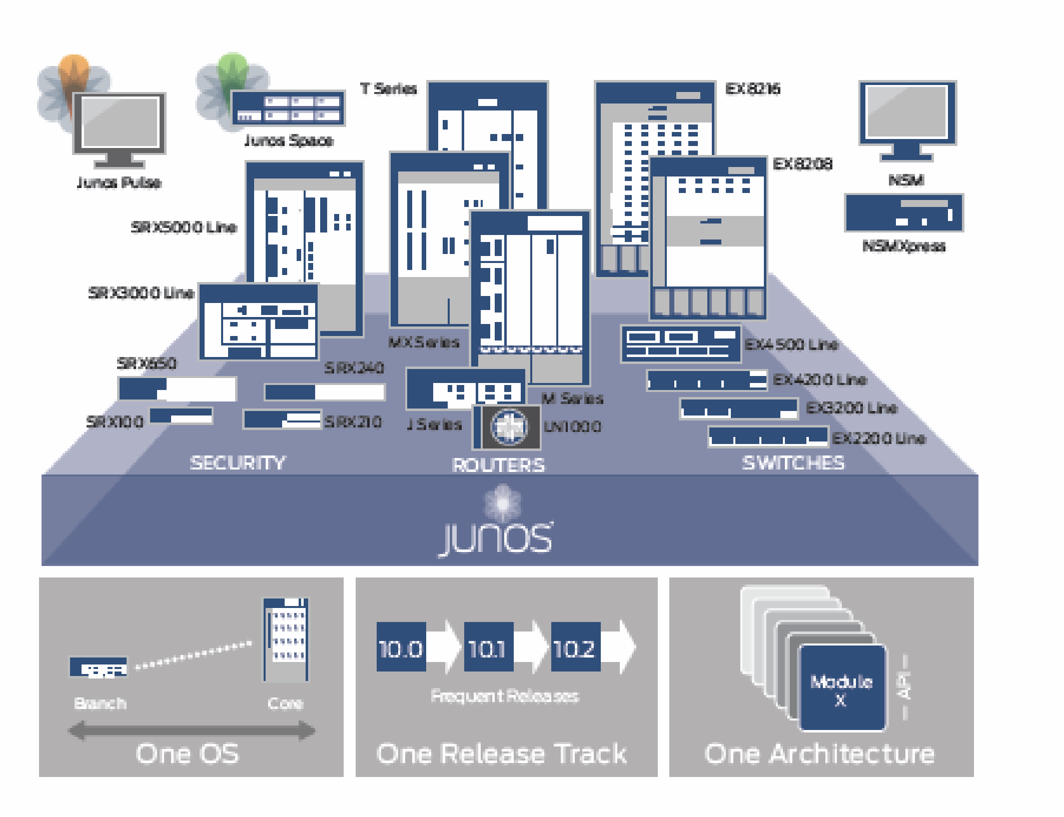
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/pathway-pages/hardware/srx240/index.pdf
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/pathway-pages/hardware/SRX3400/index.pdf
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/pathway-pages/hardware/SRX3600/index.pdf
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/pathway-pages/hardware/SRX650/index.pdf
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/pathway-pages/hardware/SRX5800/index.pdf
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx5600/hardware-guide/book-srx5600-hw-ig.pdf
- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx550/book-srx550-hw-ig.pdf

- http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx220/srx220-hardware-guide.pdf
- http://www.juniper.net/us/en/local/pdf/datasheets/1000206-en.pdf
- http://www.juniper.net/techpubs/en_US/junos12.1/information-products/pathway-pages/software-installation-and-upgrade/software-installation-and-upgrade.pdf

B. JNI's Integrated Operating System: JUNOS.

7. Juniper's operating system, known as JUNOS, is an integrated operating system, common to many but not all Juniper products. All of the Accused Products use JUNOS. As Juniper describes it, the JUNOS operating system provides "a common language across Juniper's routing, switching, and security devices…." As an important competitive point of differentiation, Juniper designed JUNOS to be "one operating systems delivering one software release track with one modular architecture." See JUNOS OS: The Power of One Operating System, at 1. (http://www.juniper.net/us/en/local/pdf/brochures/1500059-en.pdf.)

8. As Juniper captures this graphically (JUNOS OS: The Power of One Operating System, at 2).



9. As this graphic illustrates, many JNI products run on the common operating system, JUNOS.

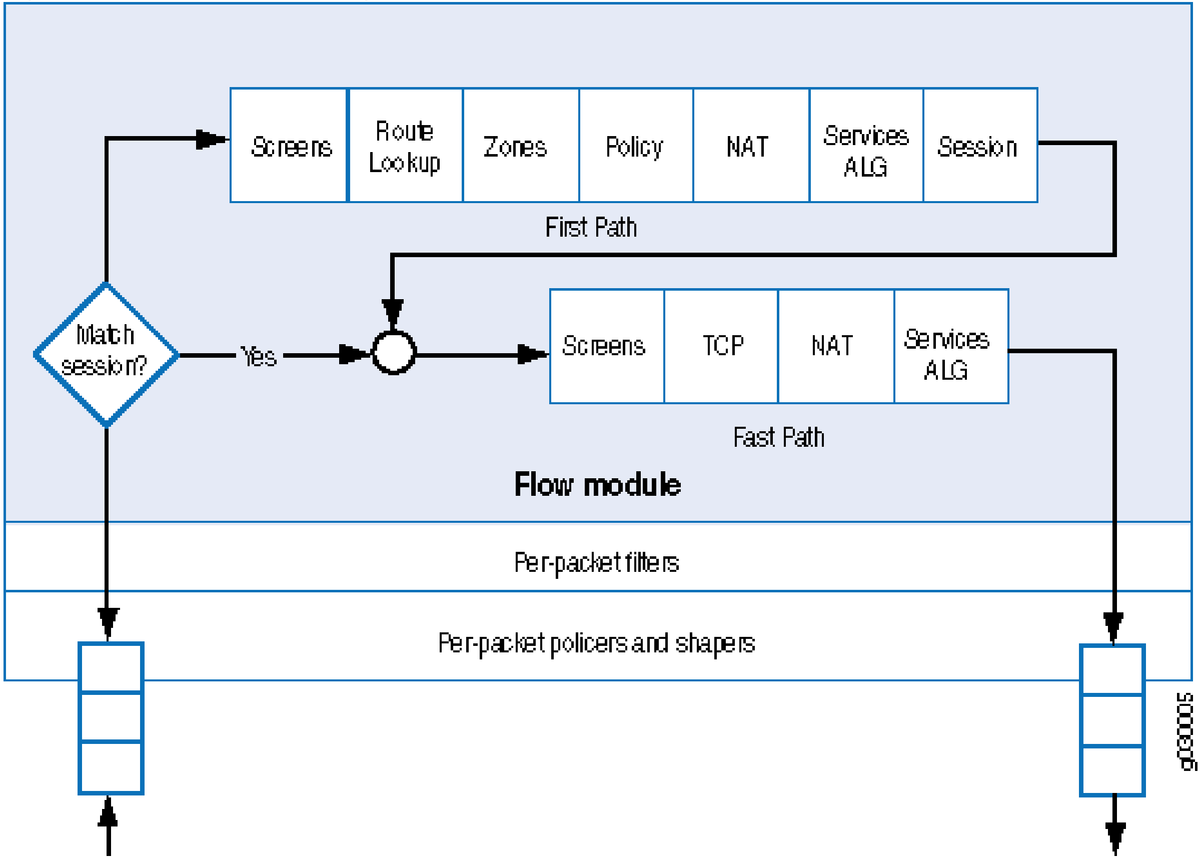
C. Flow-based Processing in JUNOS.

10. JUNOS offers flow-based (stateful) packet processing. In contrast to packet-based forwarding, where every packet stands alone without regard to flow or state information, flow-based processing requires the creation of sessions (in JNI terminology). A session is created to store, e.g., the security measures to be applied to the packets of the flow, to cache information about the state of the flow, to allocate required resources for the flow, and to provide a framework for features such as Application Layer Gateways (ALG's) and firewall features. See JUNOS Software Security Administration Guide (http://www.juniper.net/techpubs/software/junos-security/junos-security10.0/junos-security-admin-guide/junos-security-admin-guide.pdf), at 270.

11. In the flow-based functionality, JUNOS classifies the first packets of a new flow through at least a five tuple classification process. If the packet is identified as the first packet of a new flow, it goes through a policy lookup, whereby the system applies configuration information to the new packet to determine how that flow should be processed. The system then creates an instantiated and stateful data processing path, employing code modules called "plugins." The first and subsequent packets of the flow are run through this dynamically created data processing path, and state is maintained accordingly.

12. The following figure (and text) is taken from the JUNOS Software Security Configuration Guide (http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-swconfig-security/junos-security-swconfig-security.pdf) page 51 and illustrates this process. (I note similar figures and text occurs throughout Juniper's documents concerning the Accused Products.)

Figure 8: Data Packet Traversing the Flow Module on the Services Router



As a packet transits the router, it takes the following path. This packet " walk" brings together the packet-based processing and flow-based processing that JUNOS Software performs on the packet.

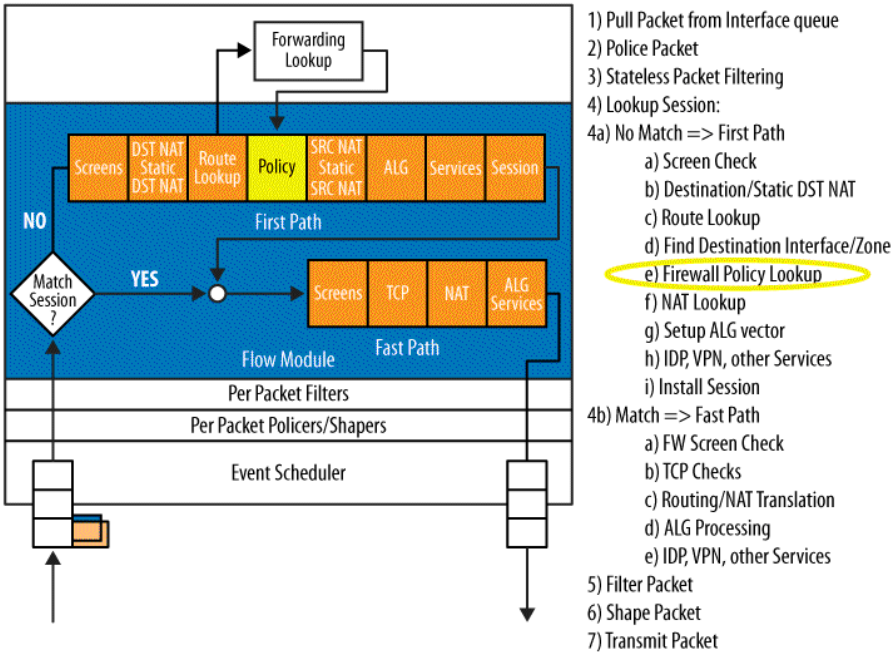
13. Pages 51-53 of the same document walks through this process in detail. Similar and related figures and discussion also appears on pages 4-5, 24-33, 36-41, 44-45 of that document.

14. This approach gives the accused JNI products the ability to provide stateful firewall functionality, ALG function, NAT, and other applications of flow based processing. See Exhibit 3 (List of Plug-Ins) and JUNOS, Feature Support Reference for SRX Series and J Series Devices (http://www.juniper.net/techpubs/en_US/junos12.1/information-products/topic-collections/security/software-all/feature-support-reference/junos-security-feature-support-guide.pdf) which describe these functions.

15. Junos Security (p. 126) says:

> As illustrated in Figure 4-1, when a packet enters the SRX, the flow daemon (flowd) performs a session lookup. It does this to see whether the packet is already part of an existing session. If the packet is part of an existing session, it takes what is referred to as the fast path. If it is not found to be part of an existing session, it goes down the slow path. The fast path has fewer steps involved in checking the packet, and as a result, it is much faster at processing the packet.

16. Figure 4-1 is familiar:



1) Pull Packet from Interface queue
2) Police Packet
3) Stateless Packet Filtering
4) Lookup Session:
4a) No Match => First Path
 a) Screen Check
 b) Destination/Static DST NAT
 c) Route Lookup
 d) Find Destination Interface/Zone
 e) Firewall Policy Lookup
 f) NAT Lookup
 g) Setup ALG vector
 h) IDP, VPN, other Services
 i) Install Session
4b) Match => Fast Path
 a) FW Screen Check
 b) TCP Checks
 c) Routing/NAT Translation
 d) ALG Processing
 e) IDP, VPN, other Services
5) Filter Packet
6) Shape Packet
7) Transmit Packet

Figure 4-1. Where policy evaluation in the SRX packet flow takes place

17. Junos Security (p. 46-55) also has an extensive discussion how the Data Center SRX products set up sessions, including the roles played by the SPC and NPU.

i. Flow-based Processing from Junos Enterprise Routing

18. Junos Enterprise Routing also has a detailed discussion of how flow-based processing is done for the Accused Products, beginning with the familiar diagram (p. 598). I have quoted extensively from this document because it is essentially a detailed roadmap showing how the Accused Products infringe. I further note that the discussion below is consistent with Junipers other documents, the deposition testimony, and the Juniper source code.

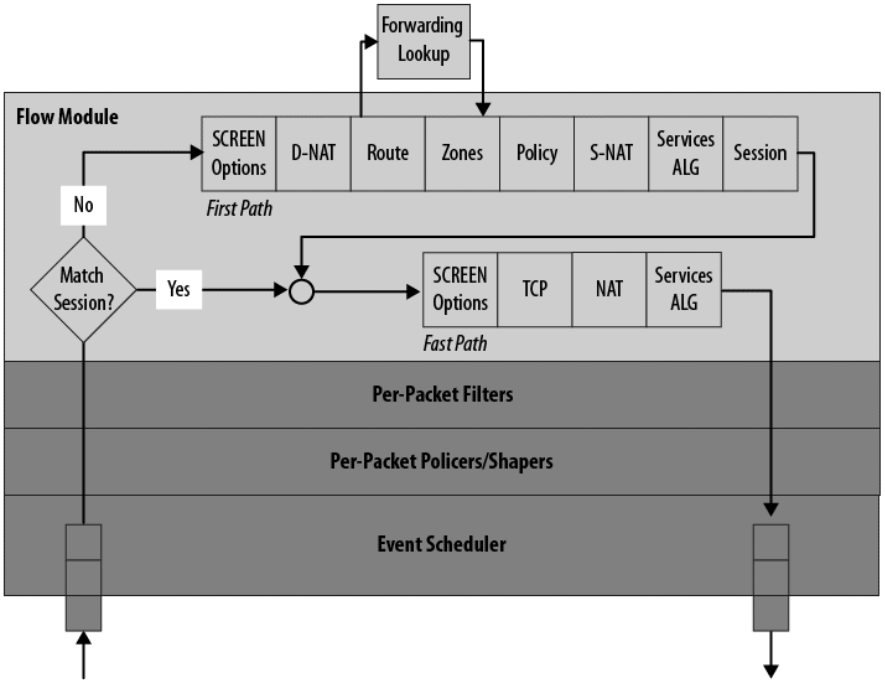


Figure 12-2. Combined packet- and flow-based processing

19. Junos Enterprise Routing states (p. 597):

> Historically, Juniper Networks routers use a packet-based forwarding model, in which each packet is individually processed and routed. In contrast, the Juniper security devices are based on a flow model. Handling traffic as flows offers significant benefits for stateful services. In the flow model, the initial packets of a communication are subjected to various levels of packet security inspections and validity checks, in addition to a single route lookup. Once the packet is deemed permissible, a corresponding session state is installed into the forwarding plane to facilitate expedited forwarding for subsequent packets belonging to the same flow. In effect, the first packets are deeply scrutinized, and the remaining packets of the same session follow a fast path through the processing.
>
> A flow is a unidirectional sequence of packets. The matching flow in the return direction is grouped to form a session, which is therefore composed of two unidirectional flows. The sessions reflect the applications that transit the firewall.

20. The final passage makes clear JNI's distinction concerning flows vs. sessions. I note that in many places Juniper documents and deponents use these closely related terms interchangeably.

21. Junos Enterprise Routing goes on to say (p. 599):

> A flow is a unidirectional stream of related packets that meet the same matching criteria and share the same characteristics. Two flows are combined (ingress and egress) to form a session. Junos treats packets belonging to the same session in the same manner. Specifically, configuration settings that determine the fate of a packet— such as the security policy that applies to it, whether the packet is sent through an IPSec tunnel, or whether NAT is applied— are assessed for the first packet of a session. The resultant set of actions and services is applied to the rest of the packets in the session. The following criteria are used to determine whether a packet matches an existing session:
>
> - Source address
> - Destination address
> - Source port
> - Destination port
> - Protocol
>
> **Flows and sessions.** The stateful handling of traffic requires the creation of a session. A session is created based on the characteristics of the first packet in a flow. Sessions are used for:
>
> - Storing security measures to be applied to the packets of the flow
> - Caching information about the state of the flow— that is, logging and counting data for a flow is cached in its session
> - Allocating required resources for features such as NAT and IPSec tunnels Providing a framework for features such as Application Layer Gateways (ALGs)
>
> The combined effects of flow and session state bring together the following features and events that affect a packet as it undergoes flow-based processing:
>
> - Flow-based forwarding
> - Session management, including session aging and changes in routes, policy, and interfaces
> - Management of VPNs, ALGs, and authentication
> - Management of policies, NAT, zones, and screens

22. Junos Enterprise Routing also says (pp. 600-601) (Figure 12-2 is shown above):

> In this section, we will follow a packet as it traverses the Junos data plane, where it encounters a mix of packet- and flow-based handling steps. Figure 12-2 shows the steps described in the following text.
>
> The steps shown for the first path represent the full set of checks and service instantiations that you can perform against the initial packets of a session. In

> contrast, the fast path represents the streamlined steps executed for previously processed (and accepted) sessions. The two-stage approach provides the ability to deeply inspect initial packets, which is computationally expensive but needed for true security, while at the same time offering high throughput by switching permitted traffic based on established session state. It should be noted that not all packets need to be touched at all possible processing points. For example, NAT is optional, and when not configured, NAT processing is not evoked. The packet processing steps are as follows:
>
> 1. Accept an incoming packet, perform class of service (CoS) behavior aggregate (BA) classification, and note the ingress interface's zone for later policy lookup.
> 2. Process the packet through the ingress policer/ shaper.
> 3. Evoke the multifield CoS classification or the firewall filter.
> 4. Perform a lookup session; if no match, follow the first path:
>   a. Conduct a firewall screen check.
>   b. Perform destination NAT as required for the incoming packet.
>   c. Perform a route lookup to determine the egress interface.
>   d. Locate the destination (outgoing) zone, based on the route lookup result.
>   e. Look up and execute policy based on incoming and outgoing zones; results include permit, deny, and reject.
>   f. Allocate the source NAT address to the packet.
>   g. Set up ALGs as needed to support identified applications.
>   h. Install a session tuple for fast path processing of related packets.
> 5. If a session is matched, follow the fast path:
>   a. Monitor the traffic for screen violations.
>   b. Perform TCP checks to look for connection anomalies and match responses.
>   c. Conduct NAT translation as required.
>   d. Perform ALG processing as needed.
> 6. Whether first or fast path, perform forwarding services on the packet based on the session information.
> 7. Perform egress firewall filtering, which can evoke a policer action.
> 8. Perform egress shaping or interface-level policing; schedule and transmit the packet.

23. The discussion above is essential a roadmap to infringement by the Accused Products. Rather than duplicate this discussion in my limitation by limitation analysis, I will refer back to it in the further proof below. To be clear, the discussion here about a limitation being met are to be taken in the context of my limitation-by-limitation discussion. As discussed

below, I use Claim 1 of the ’163 patent as an exemplar to which I refer in the analysis of the other claims. I also note that my analysis below is consistent with the Court’s claim construction.

24. **‘163 Claim 1pre:** The classification above, which involves flows and session along with the fact that the Accused Products process packets, shows this limitation is met.

25. **‘163 Claim 1a:** The discussion above concerning what sessions and flows are used for, as well as the diagram and text discussing the diagram, shows components are deployed by Juniper during flow processing.

26. **‘163 Claim 1b and ‘163 Claim 1c:** The diagram and discussion of the first path processing shows these limitations are met. Also, the resource allocation mentioned above establishes that per message per component state is being allocated.

27. **‘163 Claim 1d:** The diagram and discussion of the first path processing also shows this limitation is met. In particular “Install a session tuple for fast path processing of related packets[.]” reflects that an indication has been stored that allows the fast-path to be taken on subsequent packets of the message.

28. **‘163 Claim 1e, 1f, and 1g:** The diagram and discussion of the fast path and general Juniper discussion above shows that these limitations are met.

29. I also note that for other documents, cited above, contain similar diagrams and discussion. I do add additional lengthy and duplicative excepts here, but do expect to utilize those documents in my testimony.

ii. Flow-based Processing based on Junos source code

30. There is also a wealth of support consistent with the above Juniper documents from the Junos source code, which I will reference below in the limitation-by-limitation proof.

Redacted

Redacted

33. In JUNOS flow-based configurations, the treatment of the packets in the flow depends on the characteristics for the first packet in that flow. That is, the what of it, the content, drives the how of it, the processing.

Redacted

Redacted

Redacted

Redacted

Redacted

Redacted

ELEMENTS SECTION

51. As discussed herein, in my opinion, JNI's accused products meet the limitations of all the asserted claims of the patents-in-suit. My infringement analysis is provided below, in the JNI technology overview section, and throughout this report.

52. The evidence and reasoning as to why Claim 1 of the '163 patent is infringed applies broadly to the other asserted claims. Thus, in the analysis that follows, I have used Claim 1 of the '163 patent as an exemplar to which I refer in the analysis of the other claims. Where different or additional evidence is required to demonstrate that the other claims are meet, it has been provided as appropriate.

53. I have endeavored to provide precise references to the evidence found in Claim 1 of the '163 from the other claims. However, in general much of the evidence has broad applicability and so I expect to apply it as suitable, even if it is not explicitly referenced.

Claim 1 of the '163 patent

54. The text for claim 1 is:

1. A method in a computer system for processing a message having a sequence of packets,

the method comprising: providing a plurality of components, each component being a software routine for converting data with an input format into data with an output format;

for the first packet of the message, dynamically identifying a non-predefined sequence of components for processing the packets of the message such that the output format of the components of the non-predefined sequence match the input format of the next component in the non-predefined sequence,

wherein dynamically identifying includes selecting individual components to create the non-predefined sequence of components after the first packet is received;

and storing an indication of each of the identified components so that the non-predefined sequence does not need to be re-identified for subsequent packets of the message;

and for each of a plurality of packets of the message in sequence, for each of a plurality of components in the identified non-predefined sequence, retrieving state information relating to performing the processing of the component with the previous packet of the message;

performing the processing of the identified component with the packet and the retrieved state information;

and storing state information relating to the processing of the component with packet for use when processing the next packet of the message.

55. It is my opinion that manufacture, sale, offering for sale, or use of JNI's accused products meets the limitations of the asserted '163 patent claims, including claim 1 of the '163 patent. My infringement analysis of this claim is provided below and throughout this report.

A. Preamble

56. The text for the preamble is "*A method in a computer system for processing a message having a sequence of packets,*"

Claim Construction

57. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

58. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

59. It is my opinion that JNI's accused products meet the preamble of claim 1 under the Court's claim construction.

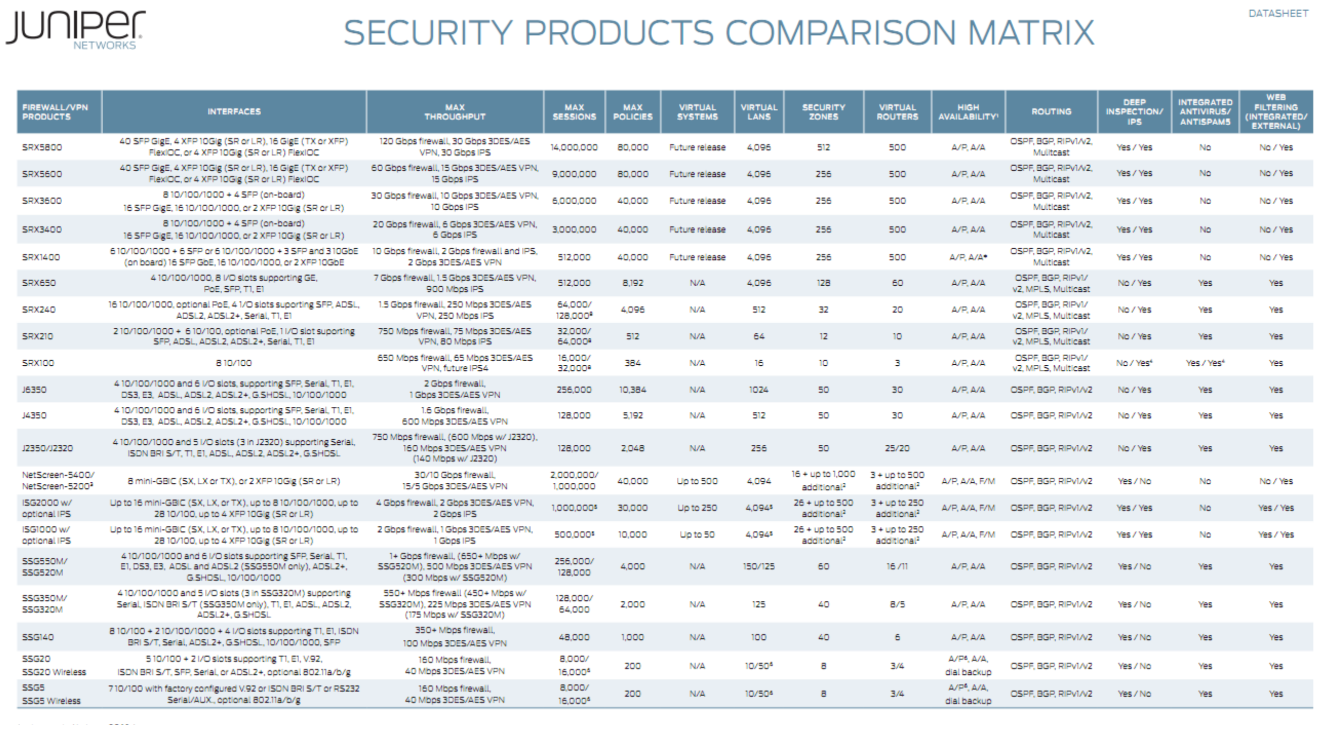
Evidence of Infringement

60. Should the Court construe the preamble to be a limitation of claim 1, in my opinion it is met. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein JNI's accused products process messages having a sequence of packets. Thus it is my opinion that JNI's accused products meet the limitation "*A method in a computer system for processing a message having a sequence of packets.*"

Redacted

62. Support for this opinion is found below in this discussion of the preamble, in the discussion of the other limitations of Claim 1, in the JNI technology overview section, and throughout this report. In the JNI technology overview section see in particular Flow-based Processing from Junos Enterprise Routing, Flow-based Processing based on Junos source code, and JNI's basic packet processing loop.

Evidence '163 C1 Pre(1)

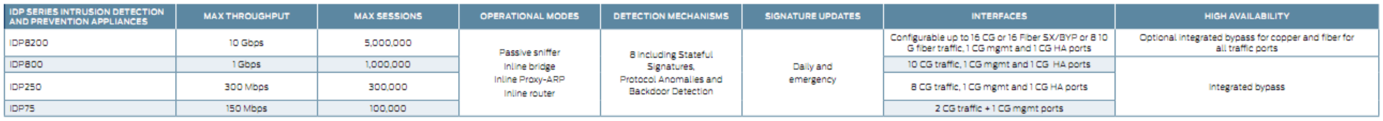


Juniper NETWORKS

DATASHEET

SECURITY PRODUCTS COMPARISON MATRIX

FIREWALL/VPN PRODUCTS	INTERFACES	MAX THROUGHPUT	MAX SESSIONS	MAX POLICIES	VIRTUAL SYSTEMS	VIRTUAL LANS	SECURITY ZONES	VIRTUAL ROUTERS	HIGH AVAILABILITY[1]	ROUTING	DEEP INSPECTION/ IPS	INTEGRATED ANTIVIRUS/ ANTISPAM5	WEB FILTERING (INTEGRATED/ EXTERNAL)
SRX5800	40 SFP GigE, 4 XFP 10Gig (SR or LR), 16 GigE (TX or XFP) FlexIOC, or 4 XFP 10Gig (SR or LR) FlexIOC	120 Gbps firewall, 30 Gbps 3DES/AES VPN, 30 Gbps IPS	14,000,000	80,000	Future release	4,096	512	500	A/P, A/A	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX5600	40 SFP GigE, 4 XFP 10Gig (SR or LR), 16 GigE (TX or XFP) FlexIOC, or 4 XFP 10Gig (SR or LR) FlexIOC	60 Gbps firewall, 15 Gbps 3DES/AES VPN, 15 Gbps IPS	9,000,000	80,000	Future release	4,096	256	500	A/P, A/A	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX3600	8 10/100/1000 + 4 SFP (on-board) 16 SFP GigE, 16 10/100/1000, or 2 XFP 10Gig (SR or LR)	30 Gbps firewall, 10 Gbps 3DES/AES VPN, 10 Gbps IPS	6,000,000	40,000	Future release	4,096	256	500	A/P, A/A	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX3400	8 10/100/1000 + 4 SFP (on-board) 16 SFP GigE, 16 10/100/1000, or 2 XFP 10Gig (SR or LR)	20 Gbps firewall, 6 Gbps 3DES/AES VPN, 6 Gbps IPS	3,000,000	40,000	Future release	4,096	256	500	A/P, A/A	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX1400	6 10/100/1000 + 6 SFP or 6 10/100/1000 + 3 SFP and 3 10GbE (on board) 16 SFP GbE, 16 10/100/1000, or 2 XFP 10GbE	10 Gbps firewall, 2 Gbps firewall and IPS, 2 Gbps 3DES/AES VPN	512,000	40,000	Future release	4,096	256	500	A/P, A/A•	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX650	4 10/100/1000, 8 I/O slots supporting GE, PoE, SFP, T1, E1	7 Gbps firewall, 1.5 Gbps 3DES/AES VPN, 900 Mbps IPS	512,000	8,192	N/A	4,096	128	60	A/P, A/A	OSPF, BGP, RIPv1/ v2, MPLS, Multicast	No / Yes	Yes	Yes
SRX240	16 10/100/1000, optional PoE, 4 I/O slots suporting SFP, ADSL, ADSL2, ADSL2+, Serial, T1, E1	1.5 Gbps firewall, 250 Mbps 3DES/AES VPN, 250 Mbps IPS	64,000/ 128,000[8]	4,096	N/A	512	32	20	A/P, A/A	OSPF, BGP, RIPv1/ v2, MPLS, Multicast	No / Yes	Yes	Yes
SRX210	2 10/100/1000 + 6 10/100, optional PoE, 1 I/O slot suporting SFP, ADSL, ADSL2, ADSL2+, Serial, T1, E1	750 Mbps firewall, 75 Mbps 3DES/AES VPN, 80 Mbps IPS	32,000/ 64,000[8]	512	N/A	64	12	10	A/P, A/A	OSPF, BGP, RIPv1/ v2, MPLS, Multicast	No / Yes	Yes	Yes
SRX100	8 10/100	650 Mbps firewall, 65 Mbps 3DES/AES VPN, future IPS4	16,000/ 32,000[8]	384	N/A	16	10	3	A/P, A/A	OSPF, BGP, RIPv1/ v2, MPLS, Multicast	No / Yes[4]	Yes / Yes[4]	Yes
J6350	4 10/100/1000 and 6 I/O slots, supporting SFP, Serial, T1, E1, DS3, E3, ADSL, ADSL2, ADSL2+, G.SHDSL, 10/100/1000	2 Gbps firewall, 1 Gbps 3DES/AES VPN	256,000	10,384	N/A	1024	50	30	A/P, A/A	OSPF, BGP, RIPv1/v2	No / Yes	Yes	Yes
J4350	4 10/100/1000 and 6 I/O slots, supporting SFP, Serial, T1, E1, DS3, E3, ADSL, ADSL2, ADSL2+, G.SHDSL, 10/100/1000	1.6 Gbps firewall, 600 Mbps 3DES/AES VPN	128,000	5,192	N/A	512	50	30	A/P, A/A	OSPF, BGP, RIPv1/v2	No / Yes	Yes	Yes
J2350/J2320	4 10/100/1000 and 5 I/O slots (3 in J2320) supporting Serial, ISDN BRI S/T, T1, E1, ADSL, ADSL2, ADSL2+, G.SHDSL	750 Mbps firewall, (600 Mbps w/ J2320), 160 Mbps 3DES/AES VPN (140 Mbps w/ J2320)	128,000	2,048	N/A	256	50	25/20	A/P, A/A	OSPF, BGP, RIPv1/v2	No / Yes	Yes	Yes
NetScreen-5400/ NetScreen-5200[3]	8 mini-GBIC (SX, LX or TX), or 2 XFP 10Gig (SR or LR)	30/10 Gbps firewall, 15/5 Gbps 3DES/AES VPN	2,000,000/ 1,000,000	40,000	Up to 500	4,094	16 + up to 1,000 additional[2]	3 + up to 500 additional[2]	A/P, A/A, F/M	OSPF, BGP, RIPv1/v2	Yes / No	No	No / Yes
ISG2000 w/ optional IPS	Up to 16 mini-GBIC (SX, LX, or TX), up to 8 10/100/1000, up to 28 10/100, up to 4 XFP 10Gig (SR or LR)	4 Gbps firewall, 2 Gbps 3DES/AES VPN, 2 Gbps IPS	1,000,000[5]	30,000	Up to 250	4,094[5]	26 + up to 500 additional[2]	3 + up to 250 additional[2]	A/P, A/A, F/M	OSPF, BGP, RIPv1/v2	Yes / Yes	No	Yes / Yes
ISG1000 w/ optional IPS	Up to 16 mini-GBIC (SX, LX, or TX), up to 8 10/100/1000, up to 28 10/100, up to 4 XFP 10Gig (SR or LR)	2 Gbps firewall, 1 Gbps 3DES/AES VPN, 1 Gbps IPS	500,000[5]	10,000	Up to 50	4,094[5]	26 + up to 500 additional[2]	3 + up to 250 additional[2]	A/P, A/A, F/M	OSPF, BGP, RIPv1/v2	Yes / Yes	No	Yes / Yes
SSG550M/ SSG520M	4 10/100/1000 and 6 I/O slots supporting SFP, Serial, T1, E1, DS3, E3, ADSL and ADSL2 (SSG550M only), ADSL2+, G.SHDSL, 10/100/1000	1+ Gbps firewall, (650+ Mbps w/ SSG520M), 500 Mbps 3DES/AES VPN (300 Mbps w/ SSG520M)	256,000/ 128,000	4,000	N/A	150/125	60	16 /11	A/P, A/A	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes
SSG350M/ SSG320M	4 10/100/1000 and 5 I/O slots (3 in SSG320M) supporting Serial, ISDN BRI S/T (SSG350M only), T1, E1, ADSL, ADSL2, ADSL2+, G.SHDSL	550+ Mbps firewall (450+ Mbps w/ SSG320M), 225 Mbps 3DES/AES VPN (175 Mbps w/ SSG320M)	128,000/ 64,000	2,000	N/A	125	40	8/5	A/P, A/A	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes
SSG140	8 10/100 + 2 10/100/1000 + 4 I/O slots supporting T1, E1, ISDN BRI S/T, Serial, ADSL2+, G.SHDSL, 10/100/1000, SFP	350+ Mbps firewall, 100 Mbps 3DES/AES VPN	48,000	1,000	N/A	100	40	6	A/P, A/A	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes
SSG20 SSG20 Wireless	5 10/100 + 2 I/O slots supporting T1, E1, V.92, ISDN BRI S/T, SFP, Serial, or ADSL2+, optional 802.11a/b/g	160 Mbps firewall, 40 Mbps 3DES/AES VPN	8,000/ 16,000[4]	200	N/A	10/50[4]	8	3/4	A/P[4], A/A, dial backup	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes
SSG5 SSG5 Wireless	7 10/100 with factory configured V.92 or ISDN BRI S/T or RS232 Serial/AUX., optional 802.11a/b/g	160 Mbps firewall, 40 Mbps 3DES/AES VPN	8,000/ 16,000[4]	200	N/A	10/50[4]	8	3/4	A/P[4], A/A, dial backup	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes



IDP SERIES INTRUSION DETECTION AND PREVENTION APPLIANCES	MAX THROUGHPUT	MAX SESSIONS	OPERATIONAL MODES	DETECTION MECHANISMS	SIGNATURE UPDATES	INTERFACES	HIGH AVAILABILITY
IDP8200	10 Gbps	5,000,000	Passive sniffer Inline bridge Inline Proxy-ARP Inline router	8 including Stateful Signatures, Protocol Anomalies and Backdoor Detection	Daily and emergency	Configurable up to 16 CG or 16 Fiber SX/BYP or 8 10 G fiber traffic, 1 CG mgmt and 1 CG HA ports	Optional integrated bypass for copper and fiber for all traffic ports
IDP800	1 Gbps	1,000,000				10 CG traffic, 1 CG mgmt and 1 CG HA ports	Integrated bypass
IDP250	300 Mbps	300,000				8 CG traffic, 1 CG mgmt and 1 CG HA ports	
IDP75	150 Mbps	100,000				2 CG traffic + 1 CG mgmt ports	

Source: *Security Products Comparison Matrix*, Published by Juniper Networks, Inc., November 2010, http://www.juniper.net/us/en/local/pdf/datasheets/1000265-en.pdf

63. Juniper's SRX product, accused here, offers flow-based processing. In addition, I note that similar diagrams as the one below are found in the JNI technology overview section and repeatedly in many of the cited Juniper documents, both in reference to the accused SRX products, but also to the J-Series routers as well.

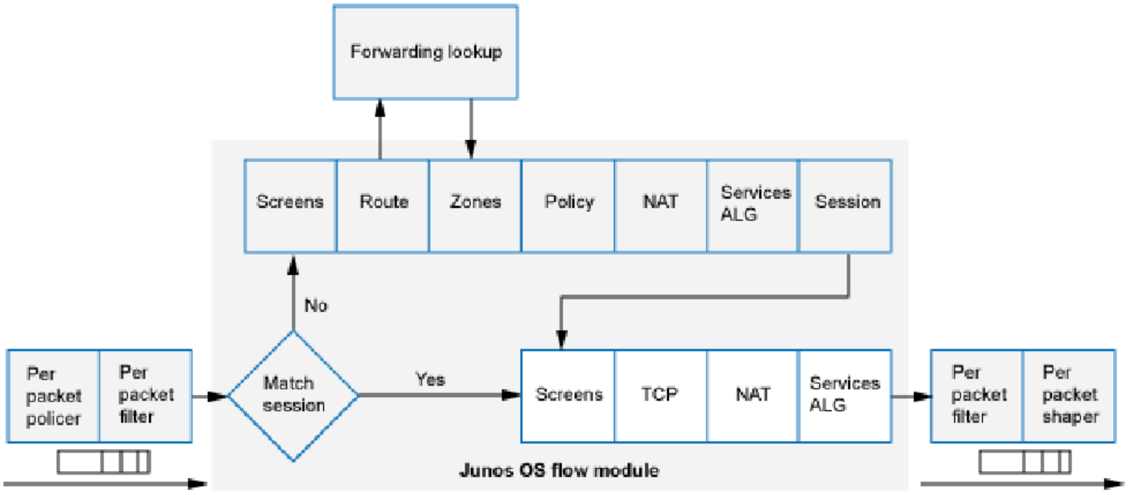
Evidence '163 C1 Pre(2)

Junos OS for SRX Series Services Gateways integrates the world-class network security and routing capabilities of Juniper Networks.

Junos OS includes a wide range of packet-based filtering, class-of-service (CoS) classifiers, and traffic-shaping features as well as a rich, extensive set of flow-based security features including policies, screens, network address translation (NAT), and other flow-based services.

Traffic that enters and exits services gateway is processed according to features you configure, such as packet filters, security policies, and screens. For example, the software can determine:

- Whether the packet is allowed into the device
- Which firewall screens to apply to the packet
- The route the packet takes to reach its destination
- Which CoS to apply to the packet, if any
- Whether to apply NAT to translate the packet's IP address
- Whether the packet requires an Application Layer Gateway (ALG)



Flow-based packet processing treats related packets, or a stream of packets, in the same way. Packet treatment depends on characteristics that were established for the first packet of the packet stream, which is referred to as a flow.

Packet-based, or stateless, packet processing treats packets discretely. Each packet is assessed individually for treatment.

A packet undergoes flow-based processing after packet-based filters and some screens have been applied to it. All flow-based processing for a single flow occurs on a single System Processing Unit (SPU). An SPU processes the packets of a flow according to the security features and other services configured for the session.

A flow is a stream of related packets that meet the same matching criteria and share the same characteristics. Junos OS treats packets belonging to the same flow in the same manner.

Source: *Junos OS Security Configuration Guide*, Published by Juniper Networks, Inc., March 2011, pages 4-5, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

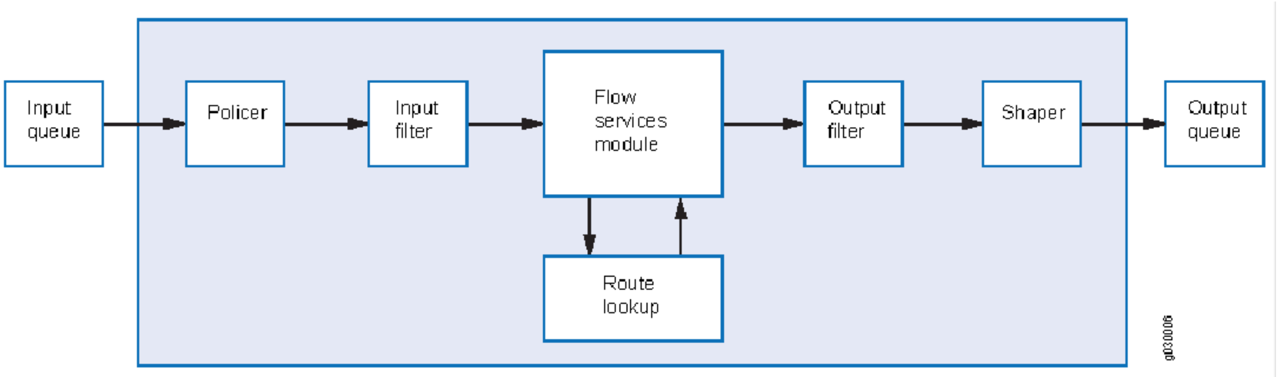
64. Junos for the J Series Routers also provides flow based processing:

Evidence '163 C1 Pre(3)

Junos OS for J Series Services Routers integrates the world-class network security and routing capabilities of Juniper Networks Operating System.

Traffic that enters and exits a services router running Junos OS is processed according to features you configure, such as security policies, packet filters, and screens. For example, the software can determine:

- Whether the packet is allowed into the router
- Which class of service (CoS) to apply to the packet, if any
- Which firewall screens to apply to the packet
- Whether to send the packet through an IPsec tunnel
- Whether the packet requires an Application Layer Gateway (ALG)
- Whether to apply Network Address Translation (NAT) to translate the packet's address
- Which route the packet uses to reach its destination



Packets that enter and exit a services router running Junos OS undergo both packet-based and flow-based processing. A device always processes packets discretely. Packet treatment depends on characteristics that were established for the first packet of the packet stream.

A packet undergoes flow-based processing after any packet-based filters and policers have been applied to it.

A flow is defined as a set of packets coming from the same source/destination addresses, source/destination ports (when applicable), protocol, and ingress/egress zones. Flows are time bound so it is possible to have packets that, while fitting the previous definition belong to different flows. For example, when an existing session is initiated and terminated, after which a new session is established using the exact same parameters as the previous session, the packets would belong to different flows.

Source: *Junos OS Security Configuration Guide*, Published by Juniper Networks, Inc., March 2011, pages 94, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

65. Thus, in my opinion and as shown above and throughout this report, the preamble is satisfied by JNI's accused products.

B. Element 1a

66. The text for 1a is "*the method comprising: providing a plurality of components, each component being a software routine for converting data with an input format into data with an output format;*."

Claim Construction

67. I understand the Court construed the term "input format" as "structure or appearance of data to be processed."

68. I understand the Court construed the term "output format" as "structure or appearance of the data that results from processing."

69. It is my opinion that JNI's accused products meet element 1a under the Court's claim construction.

Evidence of Infringement

70. It is my opinion that JNI's accused products provide a plurality of components that are software routines for converting data. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*the method comprising: providing a plurality of*

components, each component being a software routine for converting data with an input format into data with an output format;.”

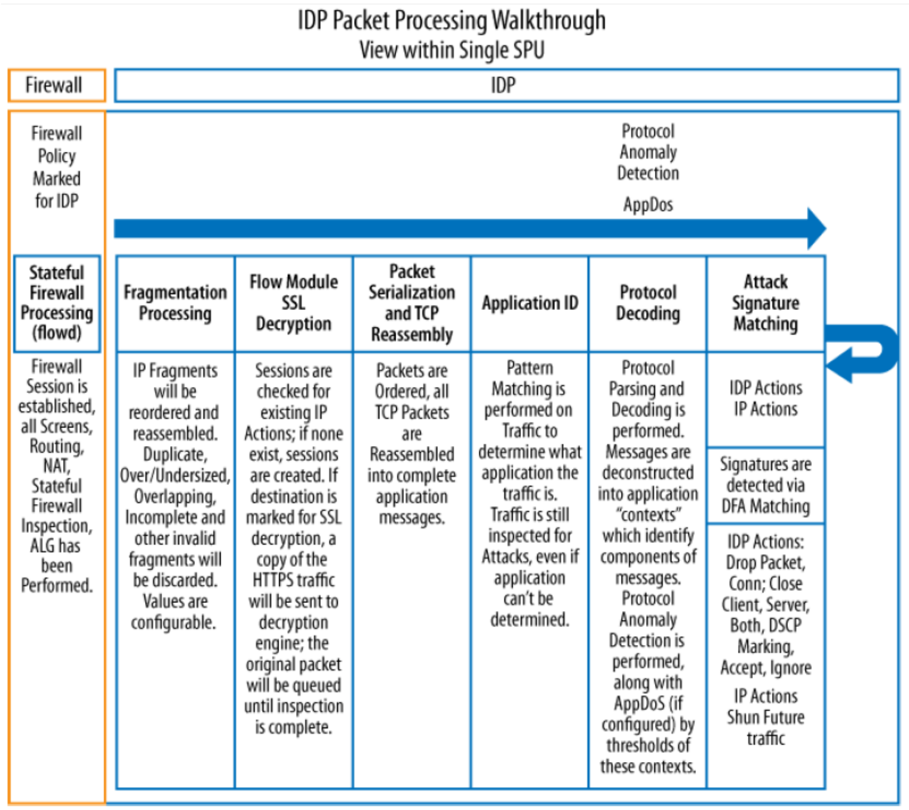
71. As discussed below, in the JNI technology overview section, and throughout this report, each module, operating at a specific network layer and performing certain processing, has an input format and an output format. In the JNI technology overview section see in particular Flow-based Processing from Junos Enterprise Routing, Flow-based Processing based on Junos source code, and JNI’s basic packet processing loop. Also see Exhibit 3 for a list of plug-ins. Thus in my opinion JNI’s accused products meet limitation 1a.

72. As described in the technical overview above, the Accused Products offer flow based processing, wherein a series of actions (modules) are instantiated as a stateful processing path, post-first packet inspection. The accused products provide components that operate on the data in sequence, with the output of one component being the input of the next. They also perform IPS algorithm processing.

73. Providing a plurality of components:

Redacted

<u>Evidence ‘163 C1 1a(2)</u>



IDP Packet Processing Walkthrough
View within Single SPU

Firewall

IDP

Firewall Policy Marked for IDP

Protocol Anomaly Detection

AppDos

Stateful Firewall Processing (flowd)

Firewall Session is established, all Screens, Routing, NAT, Stateful Firewall Inspection, ALG has been Performed.

Fragmentation Processing

IP Fragments will be reordered and reassembled. Duplicate, Over/Undersized, Overlapping, Incomplete and other invalid fragments will be discarded. Values are configurable.

Flow Module SSL Decryption

Sessions are checked for existing IP Actions; if none exist, sessions are created. If destination is marked for SSL decryption, a copy of the HTTPS traffic will be sent to decryption engine; the original packet will be queued until inspection is complete.

Packet Serialization and TCP Reassembly

Packets are Ordered, all TCP Packets are Reassembled into complete application messages.

Application ID

Pattern Matching is performed on Traffic to determine what application the traffic is. Traffic is still inspected for Attacks, even if application can't be determined.

Protocol Decoding

Protocol Parsing and Decoding is performed. Messages are deconstructed into application "contexts" which identify components of messages. Protocol Anomaly Detection is performed, along with AppDoS (if configured) by thresholds of these contexts.

Attack Signature Matching

IDP Actions
IP Actions

Signatures are detected via DFA Matching

IDP Actions: Drop Packet, Conn; Close Client, Server, Both, DSCP Marking, Accept, Ignore

IP Actions Shun Future traffic

Within the IPS engine there are several stages of processing, as illustrated above. IPS processing on the SRX can be broken down into eight general stages of processing:

Stage 1: Fragmentation processing

The first thing that must happen before you can really get to the inspection is that the SRX must process fragmented traffic (if present). To ensure that common IDS evasion techniques using fragmentation are not effective, it rebuilds any fragmented traffic from a Layer 3 perspective. This stage also provides countermeasures against fragment-based attacks such as missing fragments, underlapping or overlapping fragments, duplicate fragments, and other fragment-based anomalies. Many of these values are also configurable in the IPS sensor configuration section, although defaults should suffice in most cases.

Stage 2: IPS flow setup

After any Layer 3 fragments are processed, the SRX examines the traffic to see whether it has an existing session for it or if there is an existing session which might need some special processing. The IPS session table is different from the firewall session table, because additional IPS state related to the traffic is required.

Stage 3: SSL decryption (if applicable)

If SSL decryption is configured, and traffic is destined to a web server that is configured to be decrypted, decryption happens in this phase.

Stage 4: Serialization and reassembly

For accurate IPS processing, all messages must be processed in order, in a flow, and the messages must be reassembled if they span multiple packets. Without reassembly, an IPS engine can be easily evaded, which would result in lots of false positives. The SRX IPS engine ensures that before traffic is processed, it is ordered and reassembled in this stage of the processing.

Stage 5: Application identification

The SRX has the ability to detect what application is running on any Layer 4 port. This is important because it allows the device to determine what traffic is running in a given flow regardless of whether it is running on a standard port. Even if the application cannot be identified, the SRX can still inspect it as a bytestream. This
stage typically happens within the first couple of kilobytes of traffic, and the SRX utilizes both directions of the traffic to identify the application.

Stage 6: Protocol decoding

Once the application is identified (or is simply classified as a stream), the SRX decodes the application from a protocol level, a process known as protocol decoding. Protocol decoding allows the SRX to chop up the traffic into contexts, which are specific parts of different messages. Contexts are very important to IPS processing because they allow the SRX to look for attacks in the specific location where they actually occur, not just blindly by byte matching across all traffic that passes through the SRX. After all, you wouldn’t want the SRX to block an email conversation between you and a peer discussing the latest exploit; you would only want the SRX to block the exploit in the precise location where it actually occurs. At the time of this writing, the SRX supports almost 600 application contexts. Contexts are one of the ways that the SRX seeks to eliminate false positives. The protocol decoding stage is also where the SRX performs protocol anomaly protection and Application Distributed Denial of Service (AppDDoS) protection, both of which we will discuss later in this chapter.

Stage 7: Stateful signature detection

The attack objects that rely on signatures (rather than anomaly detection) are processed in the stateful signature stage of the device’s processing. These signatures are not blind pattern matches, but are highly accurate stateful signatures that not only match attacks within the contexts in which they occur, but also can be composed of multiple match criteria (using Boolean expressions between individual criteria). Typically, the attack signatures do not seek to detect a specific exploit, but rather protect against the vulnerability itself. This is important because attack exploits can vary, so writing signatures around a particular exploit is not a great tactic, but protecting against the actual vulnerability is much more powerful.

Stage 8: IDP/IP actions

Once an attack object in the IPS policy is matched, the SRX can execute an action on that specific session, along with actions on future sessions. The ability to execute an action on that particular session is known as an IDP action. IDP actions can be one of the following: No-Action, Drop-Packet, Drop-Connection, Close-Client, Close-Server, Close-Client-and-Server, DSCP-Marking, Recommended, or Ignore. IP actions are actions that can be enforced on future sessions. These actions include IP-Close, IP-Block, IP-Notify, and IP-Ratelimit.

Source: *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, pages 399-401.

74. When assembled by the accused products, these components implement a variety of IDP processing algorithms.

<u>Evidence ‘163 C1 1a(3)</u>

Application identification Port-independent application identification enhances both security and manageability by eliminating the need to manually and comprehensively configure application-port
mapping for the service objects and application objects used in the IDP rulebase and APE rulebase
rules. Beginning with IDP OS Release 5.1, the application identification feature can match extended
application signatures used in APE rulebase rules. Extended application signatures are also called nested application signatures. The predefined extended application signatures developed for IDP OS Release 5.1 include the most prevalent Web 2.0 applications running over HTTP.

User-defined application signatures If the predefined signatures do not address all of your use cases, you can use the NSM Object Manager to create custom application signatures.

Application policy enforcement The application policy enforcement (APE) rulebase enables you to mark, limit, or drop traffic that matches application signatures.

Application volume tracking The application volume tracking (AVT) feature leverages Profiler functionality to collect statistics about application usage.

Multimethod attack detection The IDP Series uses eight methods to detect malicious traffic.

Zero-day protection The IDP rulebase attack objects detect protocol usages that violate published RFCs. Protocol anomaly detection protects your network from undiscovered vulnerabilities.

Protocol decoding Juniper Networks Security Center (J-Security Center) provides a robust protocol detection engine that can decode more than 60 protocols and analyze and enforce proper usage in more than 500 contexts.

Recommended security policy and predefined attack objects J-Security Center provides a robust default security policy (called Recommended) and a comprehensive set of predefined attack objects (including those flagged as Recommended for various categories of attacks). The J-Security Center attack database includes more than 5500 signatures for identifying anomalies, attacks, spyware, and applications.

User-defined security policies and attack objects If you choose, you can use the default security policy or other predefined templates as a basis for your own user-defined security policy. Similarly, you can use the predefined attack objects as a basis for your own user-defined attack objects.

Active response methods J-Security Center attack objects are coded with recommended actions to take on the instant session, including drop packet, drop connection, close client, close server, and close client/server.
You can rely on these or set your own. In addition, when the IDP Series device detects an attack from a particular IP address, it can block connections from the IP address for a configurable duration of time.

Passive response methods The IDP Series supports several passive responses, including logging and TCP reset.

Traffic decryption and decapsulation The IDP Series can decrypt or decapsulate traffic and then inspect the payload. We support decryption of SSL and decapsulation of GRE, GTP, IPsec ESP NULL, and MPLS traffic.

Stateful signature The IDP rulebase attack object signatures are bound to protocol context. As a result, this detection method produces few false positives.

Protocol anomaly The IDP rulebase attack objects detect protocol usages that violate published RFCs. This method protects your network from undiscovered vulnerabilities.

Traffic anomaly The Traffic Anomalies rulebase uses heuristic rules to detect unexpected traffic patterns that might indicate reconnaissance or attacks. This method blocks distributed denial-of-service (DDoS) attacks and prevents reconnaissance activities.

Backdoor The Backdoor rulebase uses heuristic-based anomalous traffic patterns and packet analysis to detect Trojans and rootkits. These methods prevent proliferation of malware in case other security measures have been compromised.

IP spoofing The IDP Series device checks the validity of allowed addresses inside and outside the network, permitting only authentic traffic and blocking traffic with a disguised source.

Denial of service (DoS) The SYN Protector rulebase provides two, alternative methods to prevent SYN-flood attacks.

Network honeypot The IDP Series device impersonates vulnerable ports so you can track attacker reconnaissance activity.

Source: *IDP Series, Concepts and Examples Guide*, Published by: Juniper Networks, Inc., February 2011, pages 3-7, http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf]

75. Thus, in my opinion and as shown above and throughout this report, element 1a is satisfied by JNI's accused products.

C. Element 1b

76. The text for 1b is "*for the first packet of the message, dynamically identifying a non-predefined sequence of components for processing the packets of the message such that the output format of the components of the non-predefined sequence match the input format of the next component in the non-predefined sequence,.*"

Claim Construction

77. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

78. I understand the Court construed the term "non-predefined sequence of components" as "a sequence of software routines that was not identified before the first packet of a message was received."

79. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

80. I understand the Court construed the term "input format" as "structure or appearance of data to be processed."

81. I understand the Court construed the term "output format" as "structure or appearance of the data that results from processing."

82. It is my opinion that JNI's accused products meet element 1b under the Court's claim construction.

Evidence of Infringement

83. It is my opinion that JNI's accused products dynamically identify a non-predefined sequence of components for processing the packets of the message such that the output format of the components of the non-predefined sequence match the input format of the next component in the non-predefined sequence for the first packet of a message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*for the first packet of the message, dynamically identifying a non-predefined sequence of components for processing the packets of the message such that the output format of the components of the non-predefined sequence match the input format of the next component in the non-predefined sequence,.*" In particular, see below, the discussion for 1c, and the JNI technology overview section, as well as throughout this report. In the JNI technology

overview section see in particular Flow-based Processing from Junos Enterprise Routing, Flow-based Processing based on Junos source code, JNI's basic packet processing loop, and New Session: Creating a Data Processing Path Based on Information in the First Packet.

84. The first packet of a flow is dynamically identified using packet inspection. Note that packet inspection implies an analysis based on examining the packet headers and the data type of the packets. Based on that inspection, the accused products utilize a technique of "policy expressions," which are script-like directives that are loaded and re-loaded into the systems while they are running. They may be loaded and re-loaded into the systems by people, other systems or software, or both. The policies direct the system to identify the processing components and algorithms that are to be applied to the network traffic that is classified through the packet inspection.

85. The accused products identify a packet, look at the latest loaded and resolved policy expression which applies to that traffic/application flow, and then arrange a sequence of processing components to affect the policy expression directive. The output format of one processing step will match the input format of the next processing step. Fully custom traffic/application flow specifications, as well as fully custom processing components, can be dynamically loaded and re-loaded into the system as well. Because of the configurability of policy expressions, and traffic/applications specifications, there are near infinite resultant processing sequences – non-predefined – which will execute.

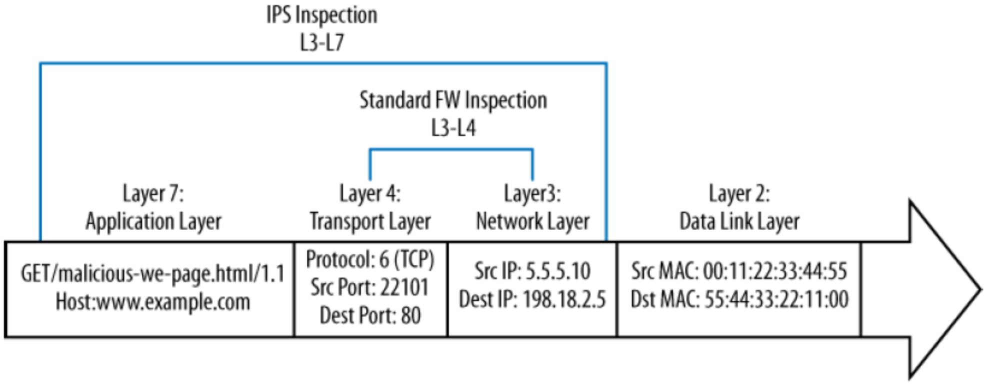
<u>Evidence '163 C1 1b(1)</u>

Identifying the First packet and matching:

Redacted

<u>Evidence '163 C1 1b(3)</u>

> At a high level, IPS works by scrutinizing all of the bits contained within packets to look for both known and unknown attacks.
>
> Traditional firewalls primarily look only at Layers 3 and 4 when it comes to security, and ignore the actual contents of the payloads themselves.



Firewall inspection of attack versus IPS

Source: *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, page 391

<u>Evidence '163 C1 1b(4)</u>

Configuration settings that determine the fate of a packet—such as the security policy that applies to it, if it requires an Application Layer Gateway (ALG), if NAT is applied to translate the packet's source and/or destination IP address—are assessed for the first packet of a flow. To determine if a flow exists for a packet, the NPU attempts to match the packet's information to that of an existing session based on the following match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique session token number for a given zone and virtual router

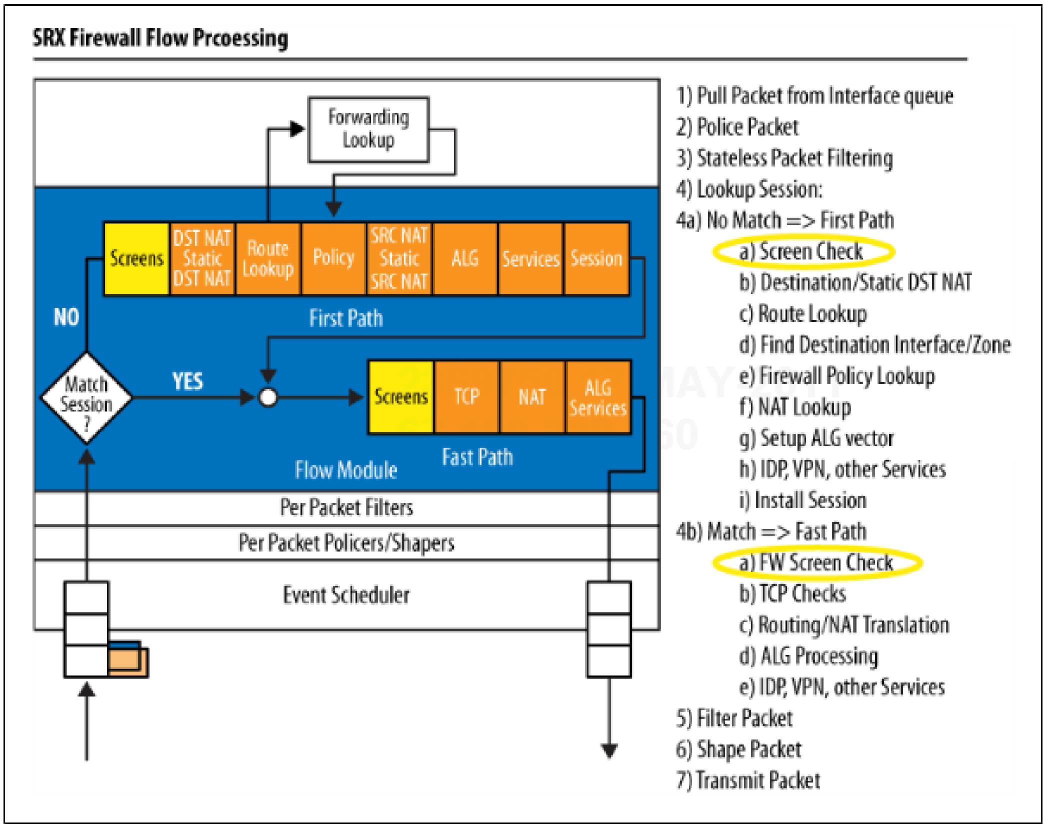
The security policy to be used for the first packet of a flow is cached in a flow table for use with the same flow and closely related flows. Security policies are associated with zones. A zone is a collection of interfaces that define a security boundary. A packet's incoming zone, as determined by the interface through which it arrived, and its outgoing zone, as determined by the forwarding lookup, together determines which policy is used for packets of the flow.

Source: *Junos OS Security Configuration Guide*, Published by Juniper Networks, Inc., March 2011, pages 5-6, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

DEEP INSPECTION

Evidence '163 C1 1b(5)

A screen is a built-in tunable protection mechanism that performs a variety of security functions to keep the network safe. Screens are extremely efficient and can be tuned to operate in a small enterprise or in the largest carrier networks. Screens are widely used to add additional protections both at the edge of the network and to internal segments to protect the network from attacks and internal misconfigurations that could impact network availability. Screens are good at detecting and preventing many types of malicious traffic. Screen checks take place very early in packet processing to make mitigation as efficient and fast as possible. Although they take more processing power than a firewall filter, they are able to look deeper into the packet and at the entire session flow, essentially enabling the SRX to block very large and complex attacks. On the higher-end SRX models, many of these screens are handled in hardware, so the traffic is dropped extremely close to the ingress interface. You may notice that the screen checks take place on both the slow path and the fast path. Once a session is permitted by policy and is established, the SRX continues to monitor that connection for signs of any malicious traffic or flooding beyond its preconfigured thresholds. If it sees any malicious traffic, it blocks and drops the packets.



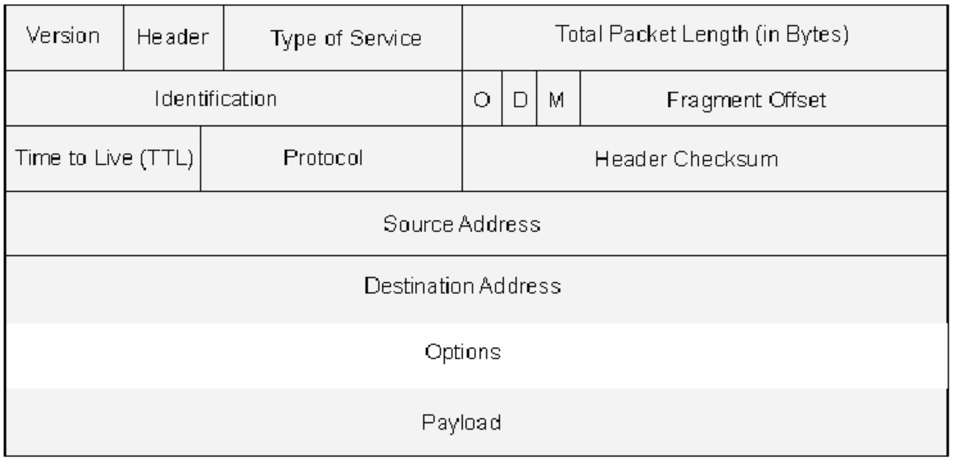
1) Pull Packet from Interface queue
2) Police Packet
3) Stateless Packet Filtering
4) Lookup Session:
4a) No Match => First Path
a) Screen Check
b) Destination/Static DST NAT
c) Route Lookup
d) Find Destination Interface/Zone
e) Firewall Policy Lookup
f) NAT Lookup
g) Setup ALG vector
h) IDP, VPN, other Services
i) Install Session
4b) Match => Fast Path
a) FW Screen Check
b) TCP Checks
c) Routing/NAT Translation
d) ALG Processing
e) IDP, VPN, other Services
5) Filter Packet
6) Shape Packet
7) Transmit Packet

Figure 7-2. Where screen checks take place in the SRX packet flow

[from *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, page 346]

Evidence '163 C1 1b(6)

IETF RFC 791 states that these options are "unnecessary for the most common communications" and, in reality, they rarely appear in IP packet headers. These options appear after the destination address in an IP packet header, as shown (here). When they do appear, they are frequently being put to some illegitimate use:



<table>
<tr><td>Version</td><td colspan="2">Header</td><td>Type of Service</td><td colspan="4">Total Packet Length (in Bytes)</td></tr>
<tr><td colspan="4">Identification</td><td>O</td><td>D</td><td>M</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live (TTL)</td><td colspan="2">Protocol</td><td colspan="4">Header Checksum</td></tr>
<tr><td colspan="8">Source Address</td></tr>
<tr><td colspan="8">Destination Address</td></tr>
<tr><td colspan="8">Options</td></tr>
<tr><td colspan="8">Payload</td></tr>
</table>

If a packet with any of the previous IP options is received, Junos OS flags this as a network reconnaissance attack and records the event for the ingress interface.

This example shows how to detect packets that use IP screen options for reconnaissance.

Source: *Junos OS Security Configuration Guide*, Published by Juniper Networks, May 2010, Page 716-717, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

86. The accused products include a generalized mechanism for doing packet inspection/flow classification.

Evidence '163 C1 1b(7)

Application identification supports user-defined custom application signatures for applications and nested applications. With custom application signatures, you can create signatures that will detect applications that are not part of the predefined application package.

Source: *Junos OS Security Configuration Guide, Juniper Networks, March 2011, Page 780-785,* https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

87. The accused products also support inspection/classification of encapsulated or encrypted traffic.

Evidence '163 C1 1b(8)

Generic Routing Encapsulation (GRE) is a tunneling protocol designed to encapsulate a wide variety of network layer packets inside IP tunneling packets. The original packet is the payload for the final packet. The protocol is used on the Internet to secure virtual private networks. To inspect the payload of an encapsulated packet, the IDP process engine must decapsulate it. IDP Series devices support decapsulation for IP-in-GRE and PPP-in-GRE.

GPRS Tunneling Protocol (or GTP) is an IP-based protocol used within Global System for Mobile communication (GSM) and Universal Mobile Telecommunications System (UMTS) networks. To inspect the payload of an encapsulated traffic, the IDP process engine must decapsulate it. IDP Series devices support decapsulation for UDP GTPv0 and GTPv1.

Internet Protocol Security (IPsec) virtual private networks use the Encapsulated Security Payload (ESP) protocol and the NULL encryption algorithm to ensure the authenticity, integrity, and confidentiality of IP packets. To inspect the payload of an encapsulated packet, the IDP process engine must decapsulate it. IDP Series devices support decapsulation for IPsec ESP NULL traffic.

Multiprotocol Label Switching (MPLS) is an IP label switching technology that enables predetermined paths to specific destinations, called Label Switched Paths (LSPs), to be established through an inherently connectionless IP network. In MPLS networks, packets contain short labels that describe how to forward them through the network. With MPLS decapsulation enabled, the IDP engine can inspect the IPv4 payload and pass through non-IPv4 payload.

Secure Sockets Layer (SSL) is a cryptographic protocol that adds security to TCP/IP communication. Several versions of the SSL and Transport Layer Security (TLS) protocols are in widespread use in applications like Web browsing, electronic mail, Internet faxing, instant messaging, and voice over IP (VoIP). SSL and TLS encrypt the Transport Layer protocol datagrams that carry the payload of these communications. While encryption is an excellent way to keep private data from prying eyes, without inspection by the IDP Series device, it also unwittingly opens a network to dangerous viruses, trojans, or network attacks. To inspect the HTTP payload of HTTPS traffic, the IDP Series device must decrypt the HTTPS session. Your security policy can examine both the SSL session and the decrypted HTTP payload.

Source: *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011*, http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, pages 177-179

Evidence '163 C1 1b(9)

To secure your network from new viruses and attacks, your security solution must offer multiple attack detection methods and an efficient way to use the various capabilities.

To stay one step ahead of these attacks, you need a solution that can adapt to ever-changing security threats and allow you to do so with minimal effort.

Juniper Networks IDP Series Intrusion Detection and Prevention Appliances with Multi-Method Detection (MMD), offers comprehensive coverage by leveraging multiple detection mechanisms. For example, by utilizing signatures, as well as other detection methods including protocol anomaly traffic anomaly detection, the Juniper Networks IDP Series appliances can thwart known attacks as well as possible future variations of the attack.

Backed by Juniper Networks Security Lab, signatures for detection of new attacks are generated on a daily basis, working very closely with many software vendors.

While an IDP solution is a critical component of every enterprise security infrastructure, it also offers the benefit of streamlining your business based on the applications used in the network. In addition to identifying viruses and attacks, the Juniper Networks IDP Series can identify the application associated with the particular traffic. Application intelligence enables accurate detection and reporting of volume used by applications such as social networking, peer-to-peer, or instant messaging. Armed with the knowledge of these applications running in the network, administrators can easily manage them by limiting bandwidth, restricting their use, or changing their prioritization for the best network optimization.

By accurately identifying and prioritizing application traffic, enterprises can ensure the necessary network bandwidth for business-critical applications without banning or blocking non-business applications. If necessary, specific application traffic can be blocked altogether to meet business or regulatory compliance.

Source: *IDP Series Intrusion Detection and Prevention Appliances*, published by Juniper Networks, Oct 2009, http://www.juniper.net/us/en/local/pdf/brochures/1500025-en.pdf]

Evidence '163 C1 1b(10)

A security policy, which can be configured from the user interface, controls the traffic flow from one zone to another zone by defining the kind(s) of traffic permitted from specified IP sources to specified IP destinations at scheduled times.

Policies allow you to **deny, permit, reject** (deny and send a TCP RST or ICMP port unreachable message to the source host), **encrypt** and **decrypt**, **authenticate, prioritize, schedule, filter**, and **monitor** the traffic attempting to cross from one security zone to another. You decide which users and what data can enter and exit, and **when and where** they can go.

Source: *Junos OS Security Configuration Guide, Juniper Networks, May 2010, Page 113,* https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

88. The accused products not only support the "firewall" types of policies mentioned above, but they support much more complicated IDP policies. IDP policies are sometimes called "rulebases" and the traffic classification specification used to match a rulebase is often called a "signature" to reflect their more general programmability.

Evidence '163 C1 1b(12)

To help block malicious application-level attacks, Juniper Networks seamlessly integrates intrusion prevention across the entire product line. For central enterprise sites, data center environments and service provider networks with high volumes of throughput, the Juniper Networks ISG Series Integrated Security Gateways with IPS, Juniper Networks SRX100, SRX210, SRX220, SRX240, SRX650, SRX1400, SRX3000 line and SRX5000 line of services gateways can be deployed for application-level protection. The ISG Series and SRX Series tightly integrates the same software found on the Juniper Networks IDP Series Intrusion Detection and Prevention Appliances to provide unmatched application-level protection against worms, trojans, spyware, and malware. More than 60 protocols are recognized including those used by advanced applications such as VoIP and streaming media.

Unmatched security processing power and network segmentation features protect critical high-speed networks against the penetration and proliferation of existing and emerging application-level threats. With multiple attack detection mechanisms, including stateful signatures and protocol anomaly, the ISG Series and SRX Series Services Gateways performs in-depth analysis of application protocol, context, state and behavior to deliver Zero-day protection.

Security administrators can deploy Juniper Networks AppSecure capability using deep inspection to block application-level attacks before they infect the network and inflict any damages. AppSecure utilizes advanced, high-performance detection mechanisms integrated with stateful inspection firewall, along with multiple threat inspection engines operating in parallel to accurately detect advanced persistent threats, including those found in nested applications within applications.

Source: *Integrated Firewall/VPN Platforms,* published by Juniper Networks, Nov. 2010, http://www.juniper.net/us/en/local/pdf/brochures/1500024-en.pdf

Evidence '163 C1 1b(13)

The IDP rulebase employs an attack object database to support two robust detection methods: stateful signatures and protocol anomalies.

A stateful signature combines an attack pattern with service, context, and other properties into a signature attack object. As a result, the IDP system does not need to expend resources inspecting huge sections of network traffic where attacks cannot possibly be, and IDP produces very few false positives.

A protocol anomaly is a deviation from protocol standards established by the Internet Engineering Taskforce (IETF) Request for Comment (RFC) process. Traffic that does not adhere to these standards is suspicious because most legitimate applications adhere to the standards, and anomalies can fairly be regarded as purposeful attempts to evade an intrusion detection system (IDS). IDP protocol-anomaly attack objects find traffic that deviates from IETF RFC standards.

When you create rules for the IDP rulebase, you specify:
- A source/destination/service match condition
- Attack objects
- Action
- Notification options

The IDP engine inspects the session beginning with the first packet to determine whether the session matches a rule. If the session matches all rule settings for source, destination, service, and VLAN tag ID, the IDP system decodes the traffic and inspects the session packets for the attack objects specified in the rule.

Source: *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011*, http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, pages 91, 92

89. Unified Threat Management (UTM) is a term used to describe the consolidation of several security features into one platform, protecting against multiple threat types.

Evidence '163 C1 1b(14)

The security features provided as part of the UTM solution are:

- **Antispam**—E-mail spam consists of unwanted e-mail messages, usually sent by commercial, malicious, or fraudulent entities. The antispam feature examines transmitted e-mail messages to identify e-mail spam. When the device detects an e-mail message deemed to be spam, it either drops the message or tags the message header or subject field with a preprogrammed string. The antispam feature uses a constantly updated spam block list (SBL). Sophos [n.b., an outside company accessed through an algorithm which goes to a special internet site], updates and maintains the IP-based SBL.

- **Full File-Based Antivirus**—A virus is executable code that infects or attaches itself to other executable code to reproduce itself. Some malicious viruses erase files or lock up systems. Other viruses merely infect files and overwhelm the target host or network with bogus data. The full file-based antivirus feature provides file-based scanning on specific Application Layer traffic checking for viruses against a virus signature database. It collects the received data packets until it has reconstructed the original application content, such as an e-mail file attachment, and then scans this content.
- **Express Antivirus**—Express antivirus scanning is offered as a less CPU intensive alternative to the full file-based antivirus feature. The express antivirus feature, like the full antivirus feature, scans specific Application Layer traffic for viruses against a virus signature database. However, unlike full antivirus, express antivirus does not reconstruct the original application content. Rather, it just sends (streams) the received data packets, as is, to the scan engine. With express antivirus, the virus scanning is executed by a hardware pattern matching engine. This improves performance while scanning is occurring, but the level of security provided is lessened.
- **Content Filtering**—Content filtering blocks or permits certain types of traffic based on the MIME type, file extension, protocol command, and embedded object type.
- **Web Filtering**—Web filtering lets you manage Internet usage by preventing access to inappropriate Web content.

Before you can configure most UTM features, you must first configure the custom objects for the feature in question. Custom objects are global parameters for UTM features. This means that configured custom objects can be applied to all UTM policies where applicable, rather than only to individual policies.

Source: *Junos OS Security Configuration Guide, Juniper Networks, May 2010, Page 585-586,* https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

Evidence ‘163 C1 1b(15)

Configuration

The unified threat management [UTM) implementation in Junos OS leverages security policies as a central point
where traffic is classified and directed to the appropriate modules for processing. In practice, a UTM policy specifying all UTM-related parameters is attached to a security policy, and

matching traffic is processed by the UTM module according to the configuration of the UTM policy.

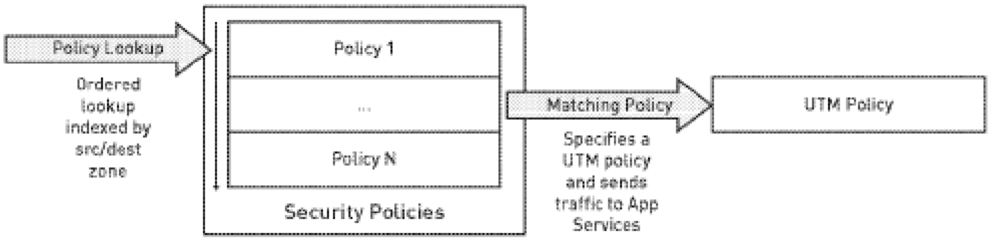

Policy Lookup
Ordered lookup indexed by src/dest zone
Policy 1
...
Policy N
Security Policies
Matching Policy
Specifies a UTM policy and sends traffic to App Services
UTM Policy

Figure 1: UTM policies

In a similar fashion, a UTM policy ties a set of protocols to one or multiple feature profiles. Each feature profile determines the specific configuration for each feature (antivirus, content filtering, anti-spam).

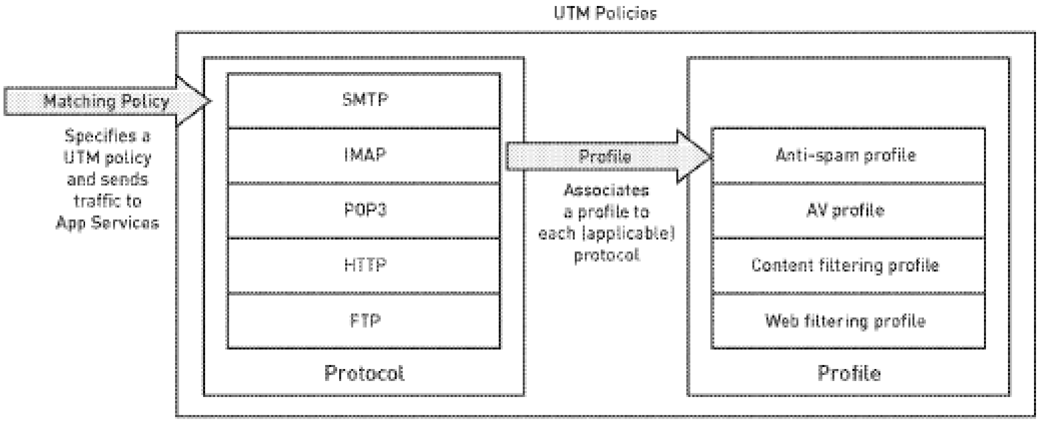

UTM Policies
Matching Policy
Specifies a UTM policy and sends traffic to App Services
SMTP
IMAP
POP3
HTTP
FTP
Protocol
Profile
Associates a profile to each (applicable) protocol
Anti-spam profile
AV profile
Content filtering profile
Web filtering profile
Profile

Figure 2: UTM policies and feature profiles

Source: "*Application Note:* Content Filtering For Branch SRX Series and J Series", JUNIPER01475161

<u>Evidence '163 C1 1b(16)</u>

> Once a security policy specifies a UTM policy, a transparent proxy processes all matching traffic and, in the case of this book's SRX devices, modifies the contents of the traffic or generate error messages back to the user. To proxy a session, an SRX device acts both as a TCP client and as a server terminating and originating a TCP session. This uses significant resources, in terms of both memory and CPU, which puts some constraints on the total number of sessions an SRX can proxy (and, in turn, the total number of concurrent sessions using UTM features). The TCP proxy code feeds a data stream to the protocol parser which, in turn, can decode the protocols supported by UTM, namely FTP, HTTP, SMTP, POP3, and IMAP. The protocol parser extracts the relevant content from each protocol and sends it to the appropriate engine for processing, all of which is depicted in Figure 9-1.

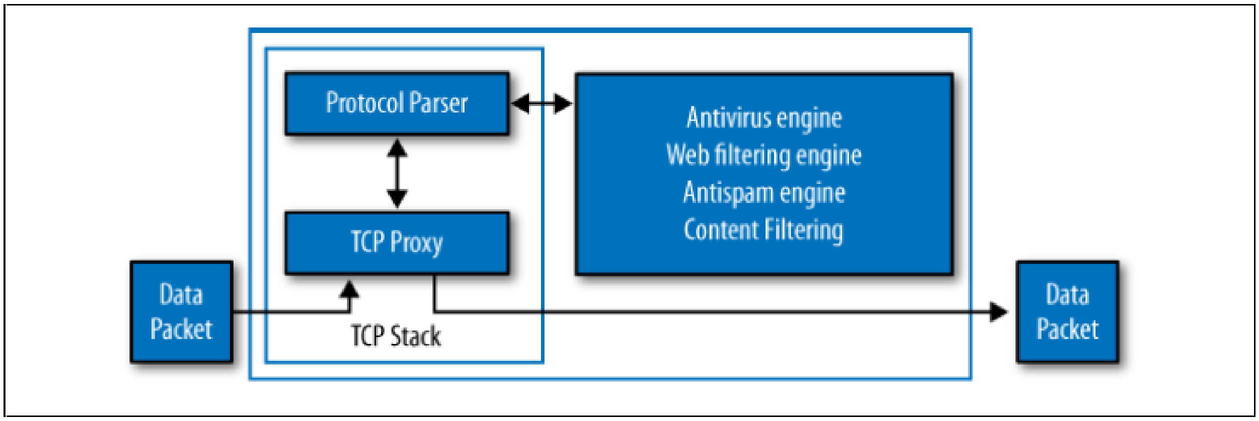


Figure 9-1. How SRX proxies a session

Source: *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, page 489

Evidence '163 C1 1b(18)

Not Just Another Chassis Design

The Dynamic Services Architecture is based on a chassis design; however, it is a complete departure from traditional chassis architecture. Rather than simply providing a fast backplane, the Dynamic Services Architecture includes the management and control necessary to incorporate individual blades into a powerful collective solution. Rather than housing disparate cards, the Dynamic Services Architecture adds each blade into a growing pool of resources. These resources can then be utilized as necessary for optimal processing of network traffic.

Switch Fabric, Control Board and Route Engine

At the heart of the Dynamic Services Architecture is the Switch fabric and Control Board (SCB). The SCB transforms the chassis from a simple blade enclosure into a highly effective mesh network. The purpose of the SCB is to allow all blades in the chassis to send traffic at extremely high bandwidth.

The Route Engine (RE) is tightly coupled with the functionality of the SCB and can be considered the central nervous system of the architecture. The RE is the control plane of the chassis and provides overall management and communications to and from system administrators, as well as calculating route tables for routing network traffic.

The operating system, which includes key chassis functionality, also runs on the RE. In the case of networking and security, functionalities such as advanced routing, switching, flow-base security, zone-based management, and screens are available on the OS.

Service Processing Cards

If the RE is the central nervous system of the chassis, the Service Processing Card (SPC), is the brain. SPCs are blades that provide the capacity to perform the heavy lifting of processing network packets.

Session Distribution

The Dynamic Services Architecture also supports automatic load balancing with advanced performance and capacity due to its session distribution design. This is enabled by the intelligent input/output and network processing subsystems, which balance sessions across the shared pool of SPCs (the “brain” discussed above). This is possible because all the SPCs in the system run the same services and have the same configuration. There is no specific mapping from one IOC to one SPC; rather, each flow is mapped dynamically upon session creation.

Any session coming in any port can be forwarded, on a session-by-session basis, to any SPC in the system.

This load balancing is performed automatically, with no configuration or oversight by the system administrator. This is dramatically opposed to traditional chassis-based solutions, where each processing blade is an independent firewall, with its own dedicated traffic, unique configuration, and routing support.

Packet Flow

In the same way, the packet flow within the Dynamic Services Architecture becomes fully integrated and far easier to manage. No longer is it necessary for administrators to provide separate instructions to each blade for traffic management. Each packet traversing the system now takes the same basic path:

1. The ingress packet enters Ethernet port on the IOC.
2. It is processed by the IOC and passed to the switch fabric.
3. One processing unit on the SPC receives and processes the packet for the firewall, IPsec VPN, and/or IPS. If the packet is to be dropped, the SPC does so and will typically log the event.
4. If the packet is to be passed, it is passed back through the switch fabric to the IOC, where it is processed by the IOC processor, where QoS is applied if necessary.
5. The packet is then passed out the Ethernet port to egress the system.

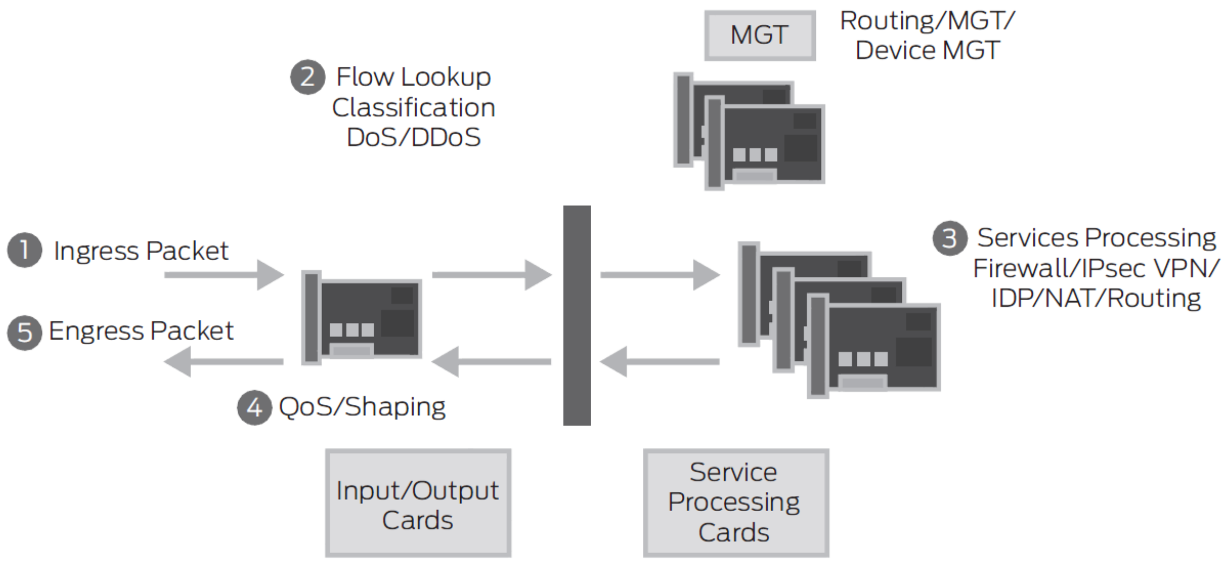


Figure 2: An example of a fully integrated packet flow (SRX5000 line)

Source: *Dynamic Services Architecture: a Revolutionary Approach to Integrated Network Security*, published by Juniper Networks, Oct 2009 , pages 4-6, https://www.juniper.net/us/en/local/pdf/whitepapers/2000288-en.pdf

90. Thus, in my opinion and as shown above and throughout this report, element 1b is satisfied by JNI's accused products.

D. Element 1c

91. The text for 1c is "*wherein dynamically identifying includes selecting individual components to create the non-predefined sequence of components after the first packet is received;.*"

Claim Construction

92. I understand the Court construed the term "selecting individual components" as "selecting the individual software routines of the sequence so that the input and output formats of the software routines are compatible."

93. I understand the Court construed the term "input format" as "structure or appearance of data to be processed."

94. I understand the Court construed the term "output format" as "structure or appearance of the data that results from processing."

95. I understand the Court construed the term "non-predefined sequence of components" as "a sequence of software routines that was not identified before the first packet of a message was received."

96. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

97. It is my opinion that JNI's accused products meet element 1c under the Court's claim construction.

Evidence of Infringement

98. It is my opinion that in JNI's accused products dynamically identifying includes selecting individual components to create the non-predefined sequence of components after the first packet is received. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*wherein dynamically identifying includes selecting individual components to create the non-predefined sequence of components after the first packet is received;.*" In particular, see below, the JNI technology overview section, 1b above, as well as throughout this report. In the JNI technology overview section see in particular Flow-based Processing from Junos Enterprise Routing, Flow-based Processing based on Junos source code, JNI's basic packet processing loop, and New Session: Creating a Data Processing Path Based on Information in the First Packet.

99. The accused products utilize a technique of "policy expression", which are script-like directives that are loaded and re-loaded into the systems while they are running. They may be loaded and re-loaded into the systems by people, other systems or software, or both. The policies direct the system to identify the processing components and algorithms which are to be applied to the network traffic which is classified through packet inspection.

100. The accused products identify a packet, look at the latest loaded and resolved policy expression which applies to that traffic/application flow, and then arrange a sequence of processing components to drive the policy expression directive. The system contains a large library of processing components. Fully custom traffic/application flow specifications can be dynamically loaded and re-loaded into the system as well.

Redacted

<u>Evidence '163 C1 1c(3)</u>

A security policy, which can be configured from the user interface, controls the traffic flow from one zone to another zone by defining the kind(s) of traffic permitted from specified IP sources to specified IP destinations at scheduled times.

Policies allow you to **deny, permit, reject** (deny and send a TCP RST or ICMP port unreachable message to the source host), **encrypt** and **decrypt**, **authenticate, prioritize, schedule, filter**, and **monitor** the traffic attempting to cross from one security zone to another. You decide which users and what data can enter and exit, and **when and where** they can go.

Source: *Junos OS Security Configuration Guide, Juniper Networks, May 2010, Page 113*, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

101. The accused products not only support the "firewall" types of policies mentioned above, but they support much more complicated IDP policies. IDP policies are sometimes called "rulebases" and the traffic classification specification used to match a rulebase is often called a "signature" to reflect their more general programmability.

Evidence '163 C1 1c(5)

To help block malicious application-level attacks, Juniper Networks seamlessly integrates intrusion prevention across the entire product line. For central enterprise sites, data center environments and service provider networks with high volumes of throughput, the Juniper Networks ISG Series Integrated Security Gateways with IPS, Juniper Networks SRX100, SRX210, SRX220, SRX240, SRX650, SRX1400, SRX3000 line and SRX5000 line of services gateways can be deployed for application-level protection. The ISG Series and SRX Series tightly integrates the same software found on the Juniper Networks IDP Series Intrusion Detection and Prevention Appliances to provide unmatched application-level protection against worms, trojans, spyware, and malware. More than 60 protocols are recognized including those used by advanced applications such as VoIP and streaming media.

Unmatched security processing power and network segmentation features protect critical high-speed networks against the penetration and proliferation of existing and emerging application-level threats. With multiple attack detection mechanisms, including stateful signatures and protocol anomaly, the ISG Series and SRX Series Services Gateways performs in-depth analysis of application protocol, context, state and behavior to deliver Zero-day protection.

Security administrators can deploy Juniper Networks AppSecure capability using deep inspection to block application-level attacks before they infect the network and inflict any damages. AppSecure utilizes advanced, high-performance detection mechanisms integrated with stateful inspection firewall, along with multiple threat inspection engines operating in parallel to accurately detect advanced persistent threats, including those found in nested applications within applications.

Source: *Integrated Firewall/VPN Platforms*, published by Juniper Networks, Nov. 2010, http://www.juniper.net/us/en/local/pdf/brochures/1500024-en.pdf

Evidence '163 C1 1c(6)

The IDP rulebase employs an attack object database to support two robust detection methods: stateful signatures and protocol anomalies.

A stateful signature combines an attack pattern with service, context, and other properties into a signature attack object. As a result, the IDP system does not need to expend resources inspecting huge sections of network traffic where attacks cannot possibly be, and IDP produces very few false positives.

A protocol anomaly is a deviation from protocol standards established by the Internet Engineering Taskforce (IETF) Request for Comment (RFC) process. Traffic that does not adhere to these standards is suspicious because most legitimate applications adhere to the standards, and anomalies can fairly be regarded as purposeful attempts to evade an intrusion detection system (IDS). IDP protocol-anomaly attack objects find traffic that deviates from IETF RFC standards.

When you create rules for the IDP rulebase, you specify:
- A source/destination/service match condition
- Attack objects
- Action
- Notification options

The IDP engine inspects the session beginning with the first packet to determine whether the session matches a rule. If the session matches all rule settings for source, destination, service, and VLAN tag ID, the IDP system decodes the traffic and inspects the session packets for the attack objects specified in the rule.

Source: *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011*, http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, pages 91, 92

102. Unified Threat Management (UTM) is a term used to describe the consolidation of several security features into one device, protecting against multiple threat types.

Evidence ‘163 C1 1c(7)

The security features provided as part of the UTM solution are:

- **Antispam**—E-mail spam consists of unwanted e-mail messages, usually sent by commercial, malicious, or fraudulent entities. The antispam feature examines transmitted e-mail messages to identify e-mail spam. When the device detects an e-mail message deemed to be spam, it either drops the message or tags the message header or subject field with a preprogrammed string. The antispam feature uses a constantly updated spam block list (SBL). Sophos [n.b., an outside company accessed through an algorithm which goes to a special internet site], updates and maintains the IP-based SBL.
- **Full File-Based Antivirus**—A virus is executable code that infects or attaches itself to other executable code to reproduce itself. Some malicious viruses erase files or lock up systems. Other viruses merely infect files and overwhelm the target host or network with bogus data. The full file-based antivirus feature provides file-based scanning on specific Application Layer traffic checking for viruses against a virus signature database. It collects the received data packets until it has reconstructed the original application content, such as an e-mail file attachment, and then scans this content.
- **Express Antivirus**—Express antivirus scanning is offered as a less CPU intensive alternative to the full file-based antivirus feature. The express antivirus feature, like the full antivirus feature, scans specific Application Layer traffic for viruses against a virus signature database. However, unlike full antivirus, express antivirus does not reconstruct the original application content. Rather, it just sends (streams) the received data packets, as is, to the scan engine. With express antivirus, the virus scanning is executed by a hardware pattern matching engine. This improves performance while scanning is occurring, but the level of security provided is lessened.
- **Content Filtering**—Content filtering blocks or permits certain types of traffic based on the MIME type, file extension, protocol command, and embedded object type.
- **Web Filtering**—Web filtering lets you manage Internet usage by preventing access to inappropriate Web content.

Before you can configure most UTM features, you must first configure the custom objects for the feature in question. Custom objects are global parameters for UTM features. This means that configured custom objects can be applied to all UTM policies where applicable, rather than only to individual policies.

Source: *Junos OS Security Configuration Guide, Juniper Networks, May 2010, Page 585-586,* https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

Evidence '163 C1 1c(8)

Once a security policy specifies a UTM policy, a transparent proxy processes all matching traffic and, in the case of this book's SRX devices, modifies the contents of the traffic or generate error messages back to the user. To proxy a session, an SRX device acts both as a TCP client and as a server terminating and originating a TCP session. This uses significant resources, in terms of both memory and CPU, which puts some constraints on the total number of sessions an SRX can proxy (and, in turn, the total number of concurrent sessions using UTM features). The TCP proxy code feeds a data stream to the protocol parser which, in turn, can decode the protocols supported by UTM, namely FTP, HTTP, SMTP, POP3, and IMAP. The protocol parser extracts the relevant content from each protocol and sends it to the appropriate engine for processing, all of which is depicted in Figure 9-1.

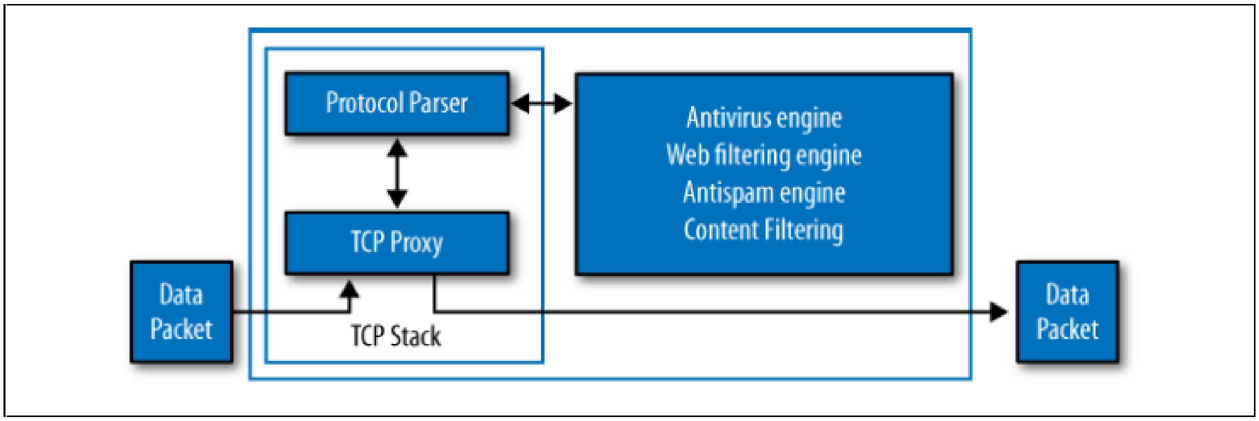


Figure 9-1. How SRX proxies a session

Source: *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, page 489

103. Juniper employs what it calls "the Dynamic Services Architecture". This architecture dynamically arranges and connects the needed components to implement the security processing identified first by the classification, and then by the policy directives.

Evidence '163 C1 1c(10)

Not Just Another Chassis Design

The Dynamic Services Architecture is based on a chassis design; however, it is a complete departure from traditional chassis architecture. Rather than simply providing a fast backplane, the Dynamic Services Architecture includes the management and control necessary to incorporate individual blades into a powerful collective solution. Rather than housing disparate cards, the Dynamic Services Architecture adds each blade into a growing pool of resources. These resources can then be utilized as necessary for optimal processing of network traffic.

Switch Fabric, Control Board and Route Engine

At the heart of the Dynamic Services Architecture is the Switch fabric and Control Board (SCB). The SCB transforms the chassis from a simple blade enclosure into a highly effective mesh network. The purpose of the SCB is to allow all blades in the chassis to send traffic at extremely high bandwidth.

The Route Engine (RE) is tightly coupled with the functionality of the SCB and can be considered the central nervous system of the architecture. The RE is the control plane of the chassis and provides overall management and communications to and from system administrators, as well as calculating route tables for routing network traffic.

The operating system, which includes key chassis functionality, also runs on the RE. In the case of networking and security, functionalities such as advanced routing, switching, flow-base security, zone-based management, and screens are available on the OS.

Service Processing Cards

If the RE is the central nervous system of the chassis, the Service Processing Card (SPC), is the brain. SPCs are blades that provide the capacity to perform the heavy lifting of processing network packets.

Session Distribution

The Dynamic Services Architecture also supports automatic load balancing with advanced performance and capacity due to its session distribution design. This is enabled by the intelligent input/output and network processing subsystems, which balance sessions across the shared pool of SPCs (the “brain” discussed above). This is possible because all the SPCs in the system run the same services and have the same configuration. There is no specific mapping from one IOC to one SPC; rather, each flow is mapped dynamically upon session creation.

Any session coming in any port can be forwarded, on a session-by-session basis, to any SPC in the system.

This load balancing is performed automatically, with no configuration or oversight by the system administrator. This is dramatically opposed to traditional chassis-based solutions, where each processing blade is an independent firewall, with its own dedicated traffic, unique configuration, and routing support.

Packet Flow

In the same way, the packet flow within the Dynamic Services Architecture becomes fully integrated and far easier to manage. No longer is it necessary for administrators to provide separate instructions to each blade for traffic management. Each packet traversing the system now takes the same basic path:

1. The ingress packet enters Ethernet port on the IOC.
2. It is processed by the IOC and passed to the switch fabric.
3. One processing unit on the SPC receives and processes the packet for the firewall, IPsec VPN, and/or IPS. If the packet is to be dropped, the SPC does so and will typically log the event.
4. If the packet is to be passed, it is passed back through the switch fabric to the IOC, where it is processed by the IOC processor, where QoS is applied if necessary.
5. The packet is then passed out the Ethernet port to egress the system.

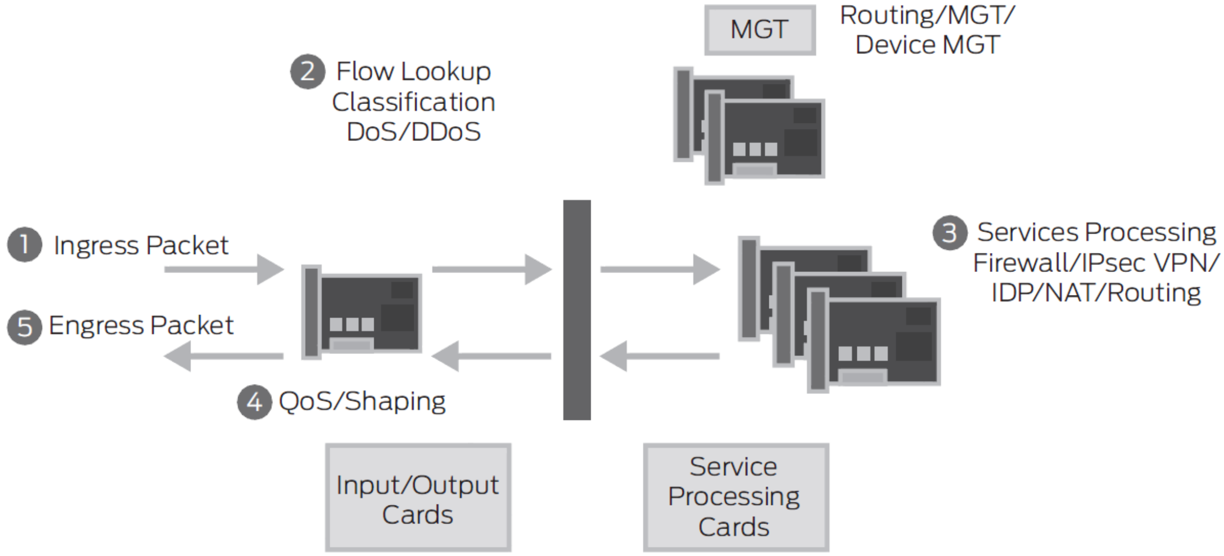


Figure 2: An example of a fully integrated packet flow (SRX5000 line)

Source: *Dynamic Services Architecture: a Revolutionary Approach to Integrated Network Security*, published by Juniper Networks, Oct 2009 , pages 4-6, https://www.juniper.net/us/en/local/pdf/whitepapers/2000288-en

104. Thus, in my opinion and as shown above and throughout this report, element 1c is satisfied by JNI's accused products.

E. Element 1d

105. The text for 1d is "*and storing an indication of each of the identified components so that the non-predefined sequence does not need to be re-identified for subsequent packets of the message;.*"

Claim Construction

106. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

107. It is my opinion that JNI's accused products meet element 1d under the Court's claim construction.

Evidence of Infringement

108. It is my opinion that for the first packet of a message JNI's accused products store an indication of each of the identified components so that the non-predefined sequence does not need to be re-identified for subsequent packets of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and storing an indication of each of the identified components so that the non-predefined sequence does not need to be re-identified for subsequent packets of the message;.*" In particular, see below, the discussion of 1g, and the JNI technology overview section, as well as throughout this report. In the JNI technology overview section see in particular Flow-

based Processing from Junos Enterprise Routing, Flow-based Processing based on Junos source code, and JNI's basic packet processing loop.

109. The accused products store information about the processing components, along with a correlation to the network traffic that those components are to operate on, as defined by the result of the non-predetermined result of the packet classification definitions, the network flows, and the executed policy directives, in accordance with this limitation. This is what the session table does. I note in particular that there is extensive evidence concerning looking up or matching of packet header information in flow tables or session tables. Generally the name of the key is not explicit, but in Flow-based Processing from Junos Enterprise Routing, it is part of the installation of a session tuple; in JNI's basic packet processing loop, it is referred to as a Redacted . In general, without such an indication, there would be no way to take the fast path indicated by so many of JNI's diagrams and in related discussion.

Redacted

Redacted

Evidence '163 C1 1d(2)

> The security policy to be used for the first packet of a flow is cached in a flow table for use with the same flow and closely related flows. Security policies are associated with zones. A zone is a collection of interfaces that define a security boundary. A packet's incoming zone, as determined by the interface through which it arrived, and its outgoing zone, as determined by the forwarding lookup, together determine which policy is used for packets of the flow.
>
> **Source:** *Junos OS Security Configuration Guide*, Published by Juniper Networks, Inc., March 2011, page 4, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

Evidence '163 C1 1d(3)

> Flow-based packet processing, which is stateful, requires the creation of sessions. A session is created for the first packet of a flow for the following purposes:
>
> • To store most of the security measures to be applied to the packets of the flow.
> • To cache information about the state of the flow.
>
> For example, logging and counting information for a flow is cached in its session. (Some stateful firewall screens rely on threshold values that pertain to individual sessions or across all sessions.)
>
> • To allocate required resources for the flow for features such as NAT.
> • To provide a framework for features such as ALGs and firewall features
>
> Most packet processing occurs in the context of a flow, including:
>
> • Management of policies, NAT, zones, and most screens.
> • Management of ALGs and authentication.
>
> **Source:** *Junos OS Security Configuration Guide*, Published by Juniper Networks, Inc., March 2011, page 6, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf]

110. Thus, in my opinion and as shown above and throughout this report, element 1d is satisfied by JNI's accused products.

F. Element 1e

111. The text for 1e is "*and for each of a plurality of packets of the message in sequence, for each of a plurality of components in the identified non-predefined sequence, retrieving state information relating to performing the processing of the component with the previous packet of the message;.*"

Claim Construction

112. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

113. I understand the Court construed the term "state information" as "information specific to a software routine for a specific message that is not information related to an overall path."

114. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

115. It is my opinion that JNI's accused products meet element 1e under the Court's claim construction.

Evidence of Infringement

116. It is my opinion that in JNI's accused products retrieve state information relating to performing the processing of the component with the previous packet of the message for each of a plurality of packets of the message in sequence and for each of a plurality of components in the identified non-predefined sequence. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and for each of a plurality of packets of the*

message in sequence, for each of a plurality of components in the identified non-predefined sequence, retrieving state information relating to performing the processing of the component with the previous packet of the message;.” See below, the JNI technology overview section, the 1f and 1g discussion. In the JNI technology overview section, see in particular Flow-based Processing from Junos Enterprise Routing, Flow-based Processing based on Junos source code, JNI’s basic packet processing loop, and Running the Code Modules (Plugins).

117. The accused products maintain state, by flow, and explained in the introductory narrative, above.

Redacted

<u>Evidence ‘163 C1 1e(2)</u>

> When the application identification feature identifies a new application, it caches the result (the destination address, port, protocol, and service) to reduce processing for subsequent sessions. The application cache and extended application cache are maintained separately.
>
> **Source:** *IDP Series Concepts and Examples Guide*, Published by Juniper Networks, Feb. 2011, Page 96, http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf

Evidence '163 C1 1e(3)

The fast-path packet process consists of the following steps:

1. An inbound packet is received by an interface and sent to the NPU, which provides processing for that interface. The NPU performs a session lookup and determines that it knows the session and the SPU processing it. The NPU then forwards the packet directly to the SPU which owns the session.
2. Policing, stateless filtering, and screens are performed. Technically, the screens that are applied after the initial packet setup are all on the NPU on the high-end SRX platforms.
3. The SPU determines if it knows about the session already, which in this case it does. The session entry will provide cached instructions on how to process the packet so that the SRX does not have to do any forwarding or policy checks, as these have already been determined in the first packet processing.

Source: *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, page 724

118. Thus, in my opinion and as shown above and throughout this report, element 1e is satisfied by JNI's accused products.

G. Element 1f

119. The text for 1f is "*performing the processing of the identified component with the packet and the retrieved state information;.*"

Claim Construction

120. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

121. I understand the Court construed the term "state information" as "information specific to a software routine for a specific message that is not information related to an overall path."

122. I understand the Court construed the term “message[s]” as “a collection of data that is related in some way, such as a stream of video or audio data or an email message.”

123. It is my opinion that JNI’s accused products meet element 1e under the Court’s claim construction.

Evidence of Infringement

124. In my opinion, JNI’s accused products perform the processing of the identified component with the packet and the retrieved state information for each of a plurality of packets of the message in sequence and for each of a plurality of components in the identified non-predefined sequence. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI’s accused products meet the limitation “*performing the processing of the identified component with the packet and the retrieved state information;.”* In particular, see below, the JNI technology overview section, the 1e and 1g discussions, as well as throughout this report. In the JNI technology overview section see in particular Flow-based Processing from Junos Enterprise Routing, Flow-based Processing based on Junos source code, JNI’s basic packet processing loop, and Running the Code Modules (Plugins).

125. The accused product performs the processing based on the retrieved state information.

Redacted

Redacted

Evidence '163 C1 1f(2)

> When the IDP engine processes security policy rules, it examines the session, beginning with the first packet, to identify a match. To match service or application, the IDP engine first compares the session against the application identification cache to identify the application. If the session does not match the application identification cache, the IDP engine processes the session against the application signatures. If the IDP engine is still unable to determine the application, it uses the standard application protocol and port.
>
> **Source:** *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011,* http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, page 96

126. Thus, in my opinion and as shown above and throughout this report, element 1f is satisfied by JNI's accused products.

H. Element 1g

127. The text for 1g is "*and storing state information relating to the processing of the component with packet for use when processing the next packet of the message."*

Claim Construction

128. I understand the Court construed the term "state information" as "information specific to a software routine for a specific message that is not information related to an overall path."

129. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

130. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

131. It is my opinion that JNI's accused products meet element 1g under the Court's claim construction.

Evidence of Infringement

132. It is my opinion that in JNI's accused products store state information relating to the processing of the component with packet for use when processing the next packet of the message for each of a plurality of packets of the message in sequence and for each of a plurality of components in the identified non-predefined sequence. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and storing state information relating to the processing of the component with packet for use when processing the next packet of the message."* In particular, see below, the JNI technology overview section, and the 1d, 1e and 1f discussions. In the JNI technology overview section see in particular Flow-based Processing from Junos Enterprise Routing, Flow-based Processing based on Junos source code, JNI's basic packet processing loop, and Running the Code Modules (Plugins).

133. State information is stored, as described in the Technical Narrative, above.

Redacted

Redacted

Evidence '163 C1 1g(2)

> Application system cache (ASC) saves the mapping between an application type and the corresponding destination IP address, destination port, protocol type, and service.
>
> Once an application is identified, its information is saved in the cache so that only one pattern matching is required for an application running on a particular system, thereby expediting the identification process. A mapping is saved in the cache only if the matched signature contains both client-to-server and server-to-client patterns. This process protects the system from hackers who might send malicious packets through a legitimate server port so that it is interpreted as a different application.
>
> By default, the application system cache saves the mapping information for 3600 seconds. However, you can configure the cache timeout value.
>
> **Source:** *Junos OS Security Configuration Guide*, Published by Juniper Networks, Inc., March 2011, page 802, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

134. Additionally, the accused products can discover their own view of the baseline security state of the network, store this state, and automatically develop security policies to detect and act on behavior which varies from that baseline.

Evidence '163 C1 1g(3)

> The Profiler is a network-analysis tool that helps you learn about your internal network so you can create effective security policies and minimize unnecessary log records. The Profiler queries and correlates information from multiple IDP Series devices.
>
> After you configure the Profiler, **it automatically learns about your internal network** and the elements that constitute it, including hosts, peers (which host is talking to which other host), ports (non-IP protocols, TCP/UDP ports, RPC programs), and Layer 7 data that uniquely identifies hosts, applications, commands, users, and filenames. You can use this data to investigate and analyze potential problems in the network and to resolve security incidents.
>
> During profiling, the IDP Series device records network activity at Layer 3, Layer 4, and Layer 7 and stores this information in a searchable database called the Profiler DB. The Profiler uses session creation, session teardown, and protocol contexts to generate this database, which defines all unique activities occurring on your network. Unique activities include attempts, probes, and successful connections. The device logs normal events only once, and it logs all unique events as often as they occur.

Baseline data gives you the building blocks for your network security policy.

After you have created a baseline and installed an appropriate security policy, you can use Profiler to alert you when new hosts or applications appear in your network. You can analyze the alerts to decide whether to update your security policy.

Source: *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011,* http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, page 32

135. Thus, in my opinion and as shown above and throughout this report, element 1g is satisfied by JNI's accused products.

I. Summary of Claim 1 of the '163 patent

136. As described above and throughout this report, in my opinion, JNI's accused products meet all the elements of claim 1 of the '163 patent.

Claim 15 of the ’163 patent

137. The text for claim 15 is:

15. A method in a computer system for demultiplexing packets of messages,

the method comprising: dynamically identifying a non-predefined sequence of components for processing each message based on the first packet of the message so that subsequent packets of the message can be processed without re-identifying the components,

wherein different non-predefined sequences of components can be identified for different messages, each component being a software routine,

and wherein dynamically identifying includes selecting individual components to create the non-predefined sequence of components;

and for each packet of each message, performing the processing of the identified non-predefined sequence of components of the message

wherein state information relating to performing the processing generated by performing the processing of a component for a packet is available to the component when the component processes the next packet of the message.

138. It is my opinion that manufacture, sale, offering for sale, or use of JNI’s accused products meets the limitations of the asserted ’163 patent claims, including claim 15 of the ’163 patent. My infringement analysis of this claim is provided below and throughout this report.

J. Preamble

139. The text for the preamble is “*A method in a computer system for demultiplexing packets of messages, “*

Claim Construction

140. I understand the Court construed the term “message[s]” as “a collection of data that is related in some way, such as a stream of video or audio data or an email message.”

141. It is my opinion that JNI's accused products meet the preamble of claim 15 under the Court's claim construction.

Evidence of Infringement

142. Should the Court construe the preamble to be a limitation of claim 15, in my opinion it is met. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein JNI's accused products demultiplex the packets of messages. Thus it is my opinion that JNI's accused products meet the limitation "*A method in a computer system for demultiplexing packets of messages,.*"

143. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the preamble section of Claim 1 of the '163 patent.

144. In addition, see the discussion of "demultiplexing" in the main body technology section.

145. Thus, in my opinion and as shown above and throughout this report, the preamble is satisfied by JNI's accused products.

K. Element 15a

146. The text for 15a is "*the method comprising: dynamically identifying a non-predefined sequence of components for processing each message based on the first packet of the message so that subsequent packets of the message can be processed without re-identifying the components,.*"

Claim Construction

147. I understand the Court construed the term "non-predefined sequence of components" as "a sequence of software routines that was not identified before the first packet of a message was received."

148. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

149. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

150. I understand the Court construed the term "based on the first packet of the message" as "relying on information in the first packet of the message."

151. It is my opinion that JNI's accused products meet element 15a under the Court's claim construction.

Evidence of Infringement

152. It is my opinion that JNI's accused products dynamically identify a non-predefined sequence of components for processing each message based on the first packet of the message so that subsequent packets of the message can be processed without re-identifying the components. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*the method comprising: dynamically identifying a non-predefined sequence of components for processing each message based on the first packet of the message so that subsequent packets of the message can be processed without re-identifying the components,.*"

153. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1a, 1b, and 1d sections of Claim 1 of the '163 patent.

154. Thus, in my opinion and as shown above and throughout this report, element 15a is satisfied by JNI's accused products.

L. Element 15b

155. The text for 15b is "*wherein different non-predefined sequences of components can be identified for different messages, each component being a software routine,.*"

Claim Construction

156. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

157. It is my opinion that JNI's accused products meet element 15b under the Court's claim construction.

Evidence of Infringement

158. It is my opinion that JNI's accused products can identify different non-predefined sequences of components for different messages, each component being a software routine. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*wherein different non-predefined sequences of components can be identified for different messages, each component being a software routine,.*"

159. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1a section of Claim 1 of the '163 patent.

160. In addition, it is clear that different messages (flows) can result in different processing. To a large extent, this follows from the basic architecture of the network. An obvious example is that TCP, UDP, and RTP are carried by IP and all need different processing. That need for different processing will result in different sequences of components being identified. Examples of this are found throughout the references cited herein. As a specific example, consider the *Junos OS Security Configuration Guide* (https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf) at pages 162-163. This is a discussion of Application Layer Gateways which considers some 14 different ALGs. Each of these ALGs would require different processing and thus result in a different sequence of components being identified.

161. Thus, in my opinion and as shown above and throughout this report, element 15b is satisfied by JNI's accused products.

M. Element 15c

162. The text for 15c is "*and wherein dynamically identifying includes selecting individual components to create the non-predefined sequence of components;.*"

Claim Construction

163. I understand the Court construed the term "selecting individual components" as "selecting the individual software routines of the sequence so that the input and output formats of the software routines are compatible."

164. I understand the Court construed the term "input format" as "structure or appearance of data to be processed."

165. I understand the Court construed the term "output format" as "structure or appearance of the data that results from processing."

166. I understand the Court construed the term "non-predefined sequence of components" as "a sequence of software routines that was not identified before the first packet of a message was received."

167. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

168. It is my opinion that JNI's accused products meet element 15c under the Court's claim construction.

Evidence of Infringement

169. It is my opinion that in JNI's accused products dynamically identifying includes selecting individual components to create the non-predefined sequence of components. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and wherein dynamically identifying includes selecting individual components to create the non-predefined sequence of components;.*"

170. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1c (and 1b) section of Claim 1 of the '163 patent.

171. Thus, in my opinion and as shown above and throughout this report, element 15c is satisfied by JNI's accused products.

N. Element 15d

172. The text for 15d is "*and for each packet of each message, performing the processing of the identified non-predefined sequence of components of the message.*"

Claim Construction

173. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

174. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

175. I understand the Court construed the term "non-predefined sequence of components" as "a sequence of software routines that was not identified before the first packet of a message was received."

176. It is my opinion that JNI's accused products meet element 15d under the Court's claim construction.

Evidence of Infringement

177. It is my opinion that for each packet of each message JNI's accused products perform the processing of the identified non-predefined sequence of components of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and for each packet of each message, performing the processing of the identified non-predefined sequence of components of the message.*"

178. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1f section of Claim 1 of the '163 patent.

179. Thus, in my opinion and as shown above and throughout this report, element 15d is satisfied by JNI's accused products.

O. Element 15e

180. The text for 15e is "*wherein state information generated by performing the processing of a component for a packet is available to the component when the component processes the next packet of the message.*"

Claim Construction

181. I understand the Court construed the term "state information" as "information specific to a software routine for a specific message that is not information related to an overall path."

182. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

183. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

184. It is my opinion that JNI's accused products meet element 15e under the Court's claim construction.

Evidence of Infringement

185. It is my opinion that in JNI's accused products state information is generated by performing the processing of a component for a packet and is available to

the component when the component processes the next packet of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*wherein state information generated by performing the processing of a component for a packet is available to the component when the component processes the next packet of the message.*"

186. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1e, 1f, and 1g sections of Claim 1 of the '163 patent.

187. Thus, in my opinion and as shown above and throughout this report, element 15e is satisfied by JNI's accused products.

P. Summary of Claim 15 of the '163 patent

188. As described above and throughout this report, in my opinion, JNI's accused products meet all the elements of claim 15 of the '163 patent.

Claim 35 of the '163 patent

189. The text for claim 35 is:

35. A computer-readable medium containing instructions for demultiplexing packets of messages,

by method comprising: dynamically identifying a message-specific non-predefined sequence of components for processing the packets of each message

upon receiving the first packet of the message wherein subsequent packets of the message can use the message-specific non-predefined sequence identified when the first packet was received,

and wherein dynamically identifying includes selecting individual components to create the message-specific non-predefined sequence of components;

and for each packet of each message, invoking the identified non-predefined sequence of components in sequence to perform the processing of each component for the packet *wherein each component saves message-specific state information so that that component can use the saved message-specific state information when the component performs its processing on the next packet of the message.*

190. It is my opinion that manufacture, sale, offering for sale, or use of JNI's accused products meets the limitations of the asserted '163 patent claims, including claim 35 of the '163 patent. My infringement analysis of this claim is provided below and throughout this report.

Q. Preamble

191. The text for the preamble is "*35. A computer-readable medium containing instructions for demultiplexing packets of messages,.*"

Claim Construction

192. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

193. It is my opinion that JNI's accused products meet the preamble of claim 35 under the Court's claim construction.

Evidence of Infringement

194. Should the Court construe the preamble to be a limitation of claim 35, in my opinion it is met. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein that JNI's accused products comprise a computer-readable medium containing instructions for demultiplexing packets of messages. Thus it is my opinion that JNI's accused products meet the limitation "*A computer-readable medium containing instructions for demultiplexing packets of messages,.*"

195. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the preamble sections of Claim 1 and Claim 15 of the ’163 patent.

196. As discussed in the JNI Hardware Section, all of the accused JNI hardware comes equipped with an internal flash drive that is used as persistent storage for the system software. The code that is cited herein when compiled are examples of said instructions. Further, this drive also satisfies the requirement for Claim 10 of the ’857 patent, because it is not a data transmission medium, and the code found on that drive comprises at least one computer-executable module. Again, the code cited herein shows examples of such modules.

197. Further, all of the accused JNI hardware contains main memory from which the system executes. Main memory is also a computer readable medium and similarly satisfies both the requirements of this claim and Claim 10 of the ’857 patent.

198. Thus, in my opinion and as shown above and throughout this report, the preamble is satisfied by JNI’s accused products.

R. Element 35a

199. The text for 35a is “*by method comprising: dynamically identifying a message-specific non-predefined sequence of components for processing the packets of each message.”*

Claim Construction

200. I understand the Court construed the term “message[s]” as “a collection of data that is related in some way, such as a stream of video or audio data or an email message.”

201. I understand the Court construed the term “non-predefined sequence of components” as “a sequence of software routines that was not identified before the first packet of a message was received.”

202. I understand the Court construed the term “processing [and variants]” as “manipulating data with a program.”

203. It is my opinion that JNI’s accused products meet element 35a under the Court’s claim construction.

Evidence of Infringement

204. It is my opinion that in JNI’s accused products said media includes instructions that dynamically identify a message-specific non-predefined sequence of components for processing the packets of each message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI’s accused products meet the limitation “*by method comprising: dynamically identifying a message-specific non-predefined sequence of components for processing the packets of each message.”*

205. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1a and 1b sections of Claim 1 of the ’163 patent.

206. Thus, in my opinion and as shown above and throughout this report, element 35a is satisfied by JNI’s accused products.

S. Element 35b

207. The text for 35b is "*upon receiving the first packet of the message wherein subsequent packets of the message can use the message-specific non-predefined sequence identified when the first packet was received,.*"

Claim Construction

208. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

209. It is my opinion that JNI's accused products meet element 35b under the Court's claim construction.

Evidence of Infringement

210. It is my opinion that in JNI's accused products said media includes instructions that upon receiving the first packet of the message arrange for subsequent packets of the message to use the message-specific non-predefined sequence identified when the first packet was received. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*upon receiving the first packet of the message wherein subsequent packets of the message can use the message-specific non-predefined sequence identified when the first packet was received,.*"

211. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1d section of Claim 1 of the '163 patent.

212. In addition, concerning "receiving," the purpose of the Accused Products is to receive network packets, process them, and, perhaps, transmit packets in response. Evidence for receiving is found throughout the citations herein. The JNI Hardware Section discussion the I/O capabilities of the Accused products.

213. In addition the packets being received are (typically) Ethernet frames carrying IP traffic supporting TCP and other IETF protocols. As such they have a set of nested headers because of encapsulation and such headers form the data type required by the '857 patent. Further details are found in the JNI technology overview section and in the main technology section.

214. Thus, in my opinion and as shown above and throughout this report, element 35b is satisfied by JNI's accused products.

T. Element 35c

215. The text for 35c is "*and wherein dynamically identifying includes selecting individual components to create the message-specific non-predefined sequence of components;.*"

Claim Construction

216. I understand the Court construed the term "selecting individual components" as "selecting the individual software routines of the sequence so that the input and output formats of the software routines are compatible."

217. I understand the Court construed the term "input format" as "structure or appearance of data to be processed."

218. I understand the Court construed the term "output format" as "structure or appearance of the data that results from processing."

219. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

220. I understand the Court construed the term "non-predefined sequence of components" as "a sequence of software routines that was not identified before the first packet of a message was received."

221. It is my opinion that JNI's accused products meet element 35c under the Court's claim construction.

Evidence of Infringement

222. It is my opinion that in JNI's accused products said media includes instructions in which dynamically identifying includes selecting individual components to create the message-specific non-predefined sequence of components. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and wherein dynamically identifying includes selecting individual components to create the message-specific non-predefined sequence of components;.*"

223. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1c (and 1b) section of Claim 1 of the '163 patent.

224. Thus, in my opinion and as shown above and throughout this report, element 35c is satisfied by JNI's accused products.

U. Element 35d

225. The text for 35d is "*and for each packet of each message, invoking the identified non-predefined sequence of components in sequence to perform the processing of each component for the packet.*"

Claim Construction

226. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

227. I understand the Court construed the term "non-predefined sequence of components" as "a sequence of software routines that was not identified before the first packet of a message was received."

228. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

229. It is my opinion that JNI's accused products meet element 35d under the Court's claim construction.

Evidence of Infringement

230. It is my opinion that, in JNI's accused products, said media includes instructions that for each packet of each message, invoke the identified non-predefined sequence of components in sequence to perform the processing of each component for the packet. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and for each packet of each message, invoking the identified non-predefined sequence of components in sequence to perform the processing of each component for the packet.*"

231. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1f section of Claim 1 of the ’163 patent.

232. Thus, in my opinion and as shown above and throughout this report, element 35d is satisfied by JNI’s accused products.

V. Element 35e

233. The text for 35e is “*wherein each component saves message-specific state information so that that component can use the saved message-specific state information when the component performs its processing on the next packet of the message.”*

Claim Construction

234. I understand the Court construed the term “message[s]” as “a collection of data that is related in some way, such as a stream of video or audio data or an email message.”

235. I understand the Court construed the term “state information” as “information specific to a software routine for a specific message that is not information related to an overall path.”

236. I understand the Court construed the term “processing [and variants]” as “manipulating data with a program.”

237. It is my opinion that JNI’s accused products meet element 35e under the Court’s claim construction.

Evidence of Infringement

238. It is my opinion that in JNI’s accused products said media includes instructions wherein each component saves message-specific state information so that

that component can use the saved message-specific state information when the component performs its processing on the next packet of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*wherein each component saves message-specific state information so that that component can use the saved message-specific state information when the component performs its processing on the next packet of the message.*"

239. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1e, 1f, and 1g sections of Claim 1 of the '163 patent.

240. Thus, in my opinion and as shown above and throughout this report, element 35e is satisfied by JNI's accused products.

W. Summary of Claim 35 of the '163 patent

241. As described above and throughout this report, in my opinion, JNI's accused products meet all the elements of claim 35 of the '163 patent.

Summary of the '163 patent

242. As discussed herein, according to my understanding JNI's accused products Systems meet the limitations of the asserted '163 patent claims 1, 15, and 35.

The ‘857 patent

243. The asserted claims of the ‘857 patent are closely related to those of the ‘163 patent. One notable difference is that in the ‘857 patent how the sequence of components is identified is more closely spelled out, either by using a “data type” or “headers.” As discussed in the JNI technology overview section in data networking, these two concepts are closely related. This relationship is also highlighted by several of the (unasserted) dependent claims of the ‘857. In fact, as further discussed in the JNI technology overview section, the accused JNI products perform the identification shown in my discussion of the ‘163 in the manner required by the asserted claims of the ‘857. I have included a discussion of the role of “data type” and “headers” in my discussion of how the limitations of the ‘163 are met. Thus below my references to the proof found in the ‘163 discussion also serves to prove the “data type” or “headers” aspects required below.

244. Another difference is that the claims of the ‘857 patent do not recite the term “non-predefined.”

Claim 1 of the ’857 patent

245. The text for claim 1 is:

1. A method in a computer system for processing packets of a message,

the method comprising: receiving a packet of the message and a data type of the message;

analyzing the data type of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence, *wherein analyzing the data type of the first packet of the message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received;*

storing an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message;

for each of a plurality of components in the identified sequence: performing the processing of each packet by the identified component;

and storing state information relating to the processing of the component with the packet for use when processing the next packet of the message.

246. It is my opinion that manufacture, sale, offering for sale, or use of JNI's accused products meets the limitations of the asserted '857 patent claims, including claim 1 of the '857 patent. My infringement analysis of this claim is provided below and throughout this report.

X. Preamble

247. The text for the preamble is "*A method in a computer system for processing packets of a message,.*"

Claim Construction

248. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

249. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

250. It is my opinion that JNI's accused products meet the preamble of claim 1 under the Court's claim construction.

Evidence of Infringement

251. Should the Court construe the preamble to be a limitation of claim 1, in my opinion it is met. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein JNI's accused products process packets of a message. Thus it is my opinion that JNI's accused products meet the limitation "*a method in a computer system for processing packets of a message,.*"

252. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the preamble section of Claim 1 of the '163 patent.

253. Thus, in my opinion and as shown above and throughout this report, the preamble is satisfied by JNI's accused products.

Y. Element 1a

254. The text for 1a is "*the method comprising: receiving a packet of the message and a data type of the message;.*"

Claim Construction

255. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

256. It is my opinion that JNI's accused products meet element 1a under the Court's claim construction.

Evidence of Infringement

257. It is my opinion that JNI's accused products receive packets of messages along with the data type of these packets. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet

the limitation "*the method comprising: receiving a packet of the message and a data type of the message;.*"

258. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 35b section of Claim 1 of the '163 patent.

259. Thus, in my opinion and as shown above and throughout this report, element 1a is satisfied by JNI's accused products.

Z. Element 1b

260. The text for 1b is "*analyzing the data type of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence,.*"

Claim Construction

261. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

262. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

263. I understand the Court construed the term "output format" as "structure or appearance of the data that results from processing."

264. I understand the Court construed the term "input format" as "structure or appearance of data to be processed."

265. It is my opinion that JNI's accused products meet element 1b under the Court's claim construction.

Evidence of Infringement

266. It is my opinion that JNI's accused products analyze the data type of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*analyzing the data type of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence,.*"

267. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1a and 1b sections of Claim 1 of the '163 patent.

268. Thus, in my opinion and as shown above and throughout this report, element 1b is satisfied by JNI's accused products.

AA. Element 1c

269. The text for 1c is "*wherein analyzing the data type of the first packet of the message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received;.*"

Claim Construction

270. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

271. I understand the Court construed the term "selecting individual components" as "selecting the individual software routines of the sequence so that the input and output formats of the software routines are compatible."

272. I understand the Court construed the term "input format" as "structure or appearance of data to be processed."

273. I understand the Court construed the term "output format" as "structure or appearance of the data that results from processing."

274. It is my opinion that JNI's accused products meet element 1c under the Court's claim construction.

Evidence of Infringement

275. It is my opinion that, in JNI's accused products, analyzing the data type of the first packet of the message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*wherein analyzing the data type of the first packet of the message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received;.*"

276. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1c (and 1b) section of Claim 1 of the ’163 patent.

277. Thus, in my opinion and as shown above and throughout this report, element 1c is satisfied by JNI’s accused products.

BB. Element 1d

278. The text for 1d is “*storing an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message;.”*

Claim Construction

279. I understand the Court construed the term “message[s]” as “a collection of data that is related in some way, such as a stream of video or audio data or an email message.”

280. It is my opinion that JNI’s accused products meet element 1d under the Court’s claim construction.

Evidence of Infringement

281. It is my opinion that JNI’s accused products store an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI’s accused products meet the limitation “*storing an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message;.”*

282. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1d section of Claim 1 of the '163 patent.

283. Thus, in my opinion and as shown above and throughout this report, element 1d is satisfied by JNI's accused products.

CC. Element 1e

284. The text for 1e is "*for each of a plurality of components in the identified sequence: performing the processing of each packet by the identified component;*"

Claim Construction

285. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

286. It is my opinion that JNI's accused products meet element 1e under the Court's claim construction.

Evidence of Infringement

287. It is my opinion that for each of a plurality of components in the identified sequence JNI's accused products perform the processing of each packet by the identified component. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*for each of a plurality of components in the identified sequence: performing the processing of each packet by the identified component;*"

288. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1f section of Claim 1 of the '163 patent.

289. Thus, in my opinion and as shown above and throughout this report, element 1e is satisfied by JNI's accused products.

DD. Element 1f

290. The text for 1e is "*and storing state information relating to the processing of the component with the packet for use when processing the next packet of the message.*"

Claim Construction

291. I understand the Court construed the term "state information" as "information specific to a software routine for a specific message that is not information related to an overall path."

292. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

293. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

294. It is my opinion that JNI's accused products meet element 1f under the Court's claim construction.

Evidence of Infringement

295. It is my opinion that for each of a plurality of components in the identified sequence JNI's accused products store state information relating to the processing of the component with the packet for use when processing the next packet of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and storing state information relating*

to the processing of the component with the packet for use when processing the next packet of the message."

296. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1e, 1f, and 1g sections of Claim 1 of the '163 patent.

297. Thus, in my opinion and as shown above and throughout this report, element 1f is satisfied by JNI's accused products.

EE. Summary of Claim 1 of the '857 patent

298. As described above and throughout this report, in my opinion, JNI's accused products meet all the elements of claim 1 of the '857 patent.

Claim 4 of the '857 patent

299. The text for claim 4 is:

4. A method in a computer system for processing a message, the message having a plurality of headers,

the method comprising: analyzing the plurality of headers of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence,

wherein analyzing the plurality of headers of the first packet of the message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received;

storing an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message;

for each of a plurality of components in the identified sequence: performing the processing of each packet by the identified component;

and storing state information relating to the processing of the component with the packet for use when processing the next packet of the message.

300. It is my opinion that manufacture, sale, offering for sale, or use of JNI's accused products meets the limitations of the asserted '857 patent claims, including claim 4 of the '857 patent. My infringement analysis of this claim is provided below and throughout this report.

FF.Preamble

301. The text for the preamble is "*A method in a computer system for processing a message, the message having a plurality of headers,.*"

Claim Construction

302. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

303. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

304. It is my opinion that JNI's accused products meet the preamble of claim 4 under the Court's claim construction.

Evidence of Infringement

305. Should the Court construe the preamble to be a limitation of claim 4, in my opinion it is met. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein JNI's accused products process messages that have a plurality of headers. Thus it is my opinion that JNI's accused products meet the limitation "*A method in a computer system for processing a message, the message having a plurality of headers,.*"

306. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the preamble section of Claim 1 of the ’163 patent.

307. Thus, in my opinion and as shown above and throughout this report, the preamble is satisfied by JNI’s accused products.

GG. Element 4a

308. The text for 4a is “*the method comprising: analyzing the plurality of headers of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence,.”*

Claim Construction

309. I understand the Court construed the term “message[s]” as “a collection of data that is related in some way, such as a stream of video or audio data or an email message.”

310. I understand the Court construed the term “processing [and variants]” as “manipulating data with a program.”

311. I understand the Court construed the term “output format” as “structure or appearance of the data that results from processing.”

312. I understand the Court construed the term “input format” as “structure or appearance of data to be processed.”

313. It is my opinion that JNI’s accused products meet element 4a under the Court’s claim construction.

Evidence of Infringement

314. It is my opinion that JNI's accused products analyze the plurality of headers of the first packet of a message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*the method comprising: analyzing the plurality of headers of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence,.*"

315. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1a and 1b sections of Claim 1 of the '163 patent.

316. Thus, in my opinion and as shown above and throughout this report, element 4a is satisfied by JNI's accused products.

HH. Element 4b

317. The text for 4b is "*wherein analyzing the plurality of headers of the first packet of the message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received;.*"

Claim Construction

318. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

319. I understand the Court construed the term "selecting individual components" as "selecting the individual software routines of the sequence so that the input and output formats of the software routines are compatible."

320. I understand the Court construed the term "input format" as "structure or appearance of data to be processed."

321. I understand the Court construed the term "output format" as "structure or appearance of the data that results from processing."

322. It is my opinion that JNI's accused products meet element 4b under the Court's claim construction.

Evidence of Infringement

323. It is my opinion that, in JNI's accused products analyzing the plurality of headers of the first packet of the message to dynamically identify the sequence of components, includes selecting individual components to form the sequence of components after the first packet of the message is received. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*wherein analyzing the plurality of headers of the first packet of the message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received;.*"

324. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1c (and 1b) section of Claim 1 of the ’163 patent.

325. Thus, in my opinion and as shown above and throughout this report, element 4b is satisfied by JNI’s accused products.

II. Element 4c

326. The text for 4c is “*storing an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message;*”

Claim Construction

327. I understand the Court construed the term “message[s]” as “a collection of data that is related in some way, such as a stream of video or audio data or an email message.”

328. It is my opinion that JNI’s accused products meet element 4c under the Court’s claim construction.

Evidence of Infringement

329. It is my opinion that JNI’s accused products store an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI’s accused products meet the limitation “*storing an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message;*”

330. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1d section of Claim 1 of the ’163 patent.

331. Thus, in my opinion and as shown above and throughout this report, element 4c is satisfied by JNI’s accused products.

JJ. Element 4d

332. The text for 4d is “*for each of a plurality of components in the identified sequence: performing the processing of each packet by the identified component;.”*

Claim Construction

333. I understand the Court construed the term “processing [and variants]” as “manipulating data with a program.”

334. It is my opinion that JNI’s accused products meet element 4d under the Court’s claim construction.

Evidence of Infringement

335. It is my opinion that for each of a plurality of components in the identified sequence JNI’s accused products perform the processing of each packet by the identified component. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI’s accused products meet the limitation “*for each of a plurality of components in the identified sequence: performing the processing of each packet by the identified component;.”*

336. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1f section of Claim 1 of the ’163 patent.

337. Thus, in my opinion and as shown above and throughout this report, element 4d is satisfied by JNI's accused products.

KK. Element 4e

338. The text for 4e is "*and storing state information relating to the processing of the component with the packet for use when processing the next packet of the message*."

Claim Construction

339. I understand the Court construed the term "state information" as "information specific to a software routine for a specific message that is not information related to an overall path."

340. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

341. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

342. It is my opinion that JNI's accused products meet element 4e under the Court's claim construction.

Evidence of Infringement

343. It is my opinion that in JNI's accused products state information relating to the processing of the component with the packet is stored for use when processing the next packet of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and*

storing state information relating to the processing of the component with the packet for use when processing the next packet of the message."

344. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1e, 1f, and 1g sections of Claim 1 of the '163 patent.

345. Thus, in my opinion and as shown above and throughout this report, element 4e is satisfied by JNI's accused products.

LL. Summary of Claim 4 of the '857 patent

346. As described above and throughout this report, in my opinion, JNI's accused products meet all the elements of claim 4 of the '857 patent.

Claim 10 of the '857 patent

347. The text for claim 10 is:

10. A computer readable storage medium, other than a data transmission medium, containing instructions for processing packets of a message, the instructions comprising at least one computer-executable module configured to:

receive a packet of the message and a data type of the message;

analyze the data type of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence,

wherein analyzing the data type of the first packet of the message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received;

store an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message;

for each of a plurality of components in the identified sequence: perform the processing of each packet by the identified component;

and store state information relating to the processing of the component with the packet for use when processing the next packet of the message.

348. It is my opinion that manufacture, sale, offering for sale, or use of JNI's accused products meets the limitations of the asserted '857 patent claims, including claim 10 of the '857 patent. My infringement analysis of this claim is provided below and throughout this report.

MM. Preamble

349. The text for the preamble is "*A computer readable storage medium, other than a data transmission medium, containing instructions for processing packets of a message, the instructions comprising at least one computer-executable module configured to:.*"

Claim Construction

350. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

351. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

352. It is my opinion that JNI's accused products meet the preamble of claim 10 under the Court's claim construction.

Evidence of Infringement

353. Should the Court construe the preamble to be a limitation of claim 10, in my opinion it is met. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein that JNI's accused products comprise a computer readable storage medium, other than a data transmission medium, that contains

instructions for processing packets of a message, the instructions comprising at least one computer-executable module configured in such a way that the other limitations of Claim 10 are met. Thus it is my opinion that JNI's accused products meet the limitation "*A computer readable storage medium, other than a data transmission medium, containing instructions for processing packets of a message, the instructions comprising at least one computer-executable module configured to:.*"

354. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the preamble section of Claim 1 and Claim 35 of the '163 patent.

355. Thus, in my opinion and as shown above and throughout this report, the preamble is satisfied by JNI's accused products.

NN. Element 10a

356. The text for 10a is "*receive a packet of the message and a data type of the message;*"

Claim Construction

357. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

358. It is my opinion that JNI's accused products meet element 10a under the Court's claim construction.

Evidence of Infringement

359. It is my opinion that in JNI's accused products said media includes at least one computer-executable module containing instructions configured to receive a packet

of the message and a data type of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*receive a packet of the message and a data type of the message;*"

360. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 35b section of Claim 1 of the '163 patent as well as the 1a section of Claim 1 of the '857 patent.

361. Thus, in my opinion and as shown above and throughout this report, element 10a is satisfied by JNI's accused products.

OO. Element 10b

362. The text for 10b is "*analyze the data type of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence,.*"

Claim Construction

363. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

364. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

365. I understand the Court construed the term "output format" as "structure or appearance of the data that results from processing."

366. I understand the Court construed the term "input format" as "structure or appearance of data to be processed."

367. It is my opinion that JNI's accused products meet element 10b under the Court's claim construction.

Evidence of Infringement

368. It is my opinion that, in JNI's accused products, said media includes at least one computer-executable module containing instructions configured to analyze the data type of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*analyze the data type of a first packet of the message to dynamically identify a sequence of components for processing a plurality of packets of the message such that the output format of the components of the sequence match the input format of the next component in the sequence,.*"

369. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1a and 1b sections of Claim 1 of the '163 patent as well as the 1b section of Claim 1 of the '857 patent.

370. Thus, in my opinion and as shown above and throughout this report, element 10b is satisfied by JNI's accused products.

PP.Element 10c

371. The text for 10c is "*wherein analyzing the data type of the first packet of the message to dynamically identify the sequence of components includes selecting*

individual components to form the sequence of components after the first packet of the message is received;.”

Claim Construction

372. I understand the Court construed the term “message[s]” as “a collection of data that is related in some way, such as a stream of video or audio data or an email message.”

373. I understand the Court construed the term “selecting individual components” as “selecting the individual software routines of the sequence so that the input and output formats of the software routines are compatible.”

374. I understand the Court construed the term “input format” as “structure or appearance of data to be processed.”

375. I understand the Court construed the term “output format” as “structure or appearance of the data that results from processing.”

376. It is my opinion that JNI’s accused products meet element 10c under the Court’s claim construction.

Evidence of Infringement

377. It is my opinion that in JNI’s accused products said media includes at least one computer-executable module containing instructions configured such that analyzing the data type of the first packet of the message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI’s accused products meet the limitation “*wherein analyzing the data type of the first packet of the*

message to dynamically identify the sequence of components includes selecting individual components to form the sequence of components after the first packet of the message is received;.”

378. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1c (and 1b) section of Claim 1 of the ’163 patent as well as the 1c section of Claim 1 of the ’857 patent.

379. Thus, in my opinion and as shown above and throughout this report, element 10c is satisfied by JNI’s accused products.

QQ. Element 10d

380. The text for 10d is “*store an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message;.”*

Claim Construction

381. I understand the Court construed the term “message[s]” as “a collection of data that is related in some way, such as a stream of video or audio data or an email message.”

382. It is my opinion that JNI’s accused products meet element 10d under the Court’s claim construction.

Evidence of Infringement

383. It is my opinion that in JNI’s accused products said media includes at least one computer-executable module containing instructions configured to store an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message. As is evident from the JNI documents,

deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*store an indication of each of the identified components so that the sequence does not need to be re-identified for subsequent packets of the message;.*"

384. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1d section of Claim 1 of the '163 patent as well as the 1d section of Claim 1 of the '857 patent.

385. Thus, in my opinion and as shown above and throughout this report, element 10d is satisfied by JNI's accused products.

RR. Element 10e

386. The text for 10e is "*for each of a plurality of components in the identified sequence: perform the processing of each packet by the identified component;.*"

Claim Construction

387. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

388. It is my opinion that JNI's accused products meet element 10e under the Court's claim construction.

Evidence of Infringement

389. It is my opinion that in JNI's accused products said media includes at least one computer-executable module containing instructions configured to for each of a plurality of components in the identified sequence: perform the processing of each packet by the identified component. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the

limitation "*for each of a plurality of components in the identified sequence: perform the processing of each packet by the identified component.*"

390. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1f section of Claim 1 of the '163 patent as well as the 1e section of Claim 1 of the '857 patent.

391. Thus, in my opinion and as shown above and throughout this report, element 10e is satisfied by JNI's accused products.

SS. Element 10f

392. The text for 10e is "*and store state information relating to the processing of the component with the packet for use when processing the next packet of the message.*"

Claim Construction

393. I understand the Court construed the term "state information" as "information specific to a software routine for a specific message that is not information related to an overall path."

394. I understand the Court construed the term "message[s]" as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

395. I understand the Court construed the term "processing [and variants]" as "manipulating data with a program."

396. It is my opinion that JNI's accused products meet element 10f under the Court's claim construction.

Evidence of Infringement

397. It is my opinion that in JNI's accused products said media includes at least one computer-executable module containing instructions configured to for each of a plurality of components in the identified sequence: store state information relating to the processing of the component with the packet for use when processing the next packet of the message. As is evident from the JNI documents, deposition testimony, code, and other evidence cited herein, JNI's accused products meet the limitation "*and store state information relating to the processing of the component with the packet for use when processing the next packet of the message.*"

398. Support is found in the evidence and discussion throughout this report and particularly in the JNI technology overview section and the 1e, 1f, and 1g sections of Claim 1 of the '163 patent as well as the 1f section of Claim 1 of the '857 patent.

399. Thus, in my opinion and as shown above and throughout this report, element 10f is satisfied by JNI's accused products.

TT. Summary of Claim 10 of the '857 patent

400. As described above and throughout this report, in my opinion, JNI's accused products meet all the elements of claim 10 of the '857 patent.

Summary of the '857 patent

401. As discussed herein, according to my understanding JNI's accused products Systems meet the limitations of the asserted '857 patent claims 1, 4, and 10.

EXHIBIT 9

UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

```
IMPLICIT NETWORKS, INC.,        )
                                )
        Plaintiff,              )
                                )
        vs.                     )   No. C 10-4234 SI
                                )
JUNIPER NETWORKS, INC.,         )
                                )
        Defendant.              )
______________________________  )
```

HIGHLY CONFIDENTIAL

30(B)(6) DEPOSITION OF: KRISHNA NARAYANASWAMY

TAKEN ON: September 20, 2011

12950 BRENDA L. MARSHALL

CSR No. 6939

Page 33

packet, to -- associated with a particular flow.

So, again, it can be -- the match criteria for flow can involve a whole set of attributes that are there in the packet.

10:16:20 Q. There's a six-tuple listed here, is there not?

A. This is an example. Let me see. I didn't count it. I see more than six tuples here. I see almost seven tuples, or it could be 10:16:34 more because there seems to be things in plural, so -- like service tokens.

Q. Okay. And more on that in a minute. Okay.

And what does the system do, once it 10:16:48 associates packets with a flow?

MR. MCPHIE: Objection. Vague and ambiguous.

THE WITNESS: Can you be more specific?

BY MR. HOSIE:

10:16:58 Q. You're confused by that question?

A. Yes, I am.

Q. All right. Let's -- let's take -- in terms of flow-based stateful processing, as used for JUNOS software for the J-series router, as 10:17:10 per this document, the document you said you

Page 34

were familiar with, tell me what happens when the first packet of a new flow hits the device, a J-series router. We're going to walk through it step by step.

10:17:24 A. Okay.

Q. You with me?

A. Yes, I'm with you.

Q. Thank you.

A. So when the first packet --

10:17:28 MR. MCPHIE: Same objection. Vague and ambiguous. Incomplete hypothetical.

THE WITNESS: So when the first packet arrives at the system, it is matched to a flow, and that is based on the configuration of the

10:17:44 box as to what identifies a flow. So it is not -- it is based on -- on configuration, it's not dynamic. It has to be configured on the box as to what a flow is.

BY MR. HOSIE:

10:17:56 Q. So some system admin has to say, "Okay, here are the definitions of the flows we are going to process"; true?

A. No. That's not true.

Q. How does it work?

10:18:08 A. So, again, as you can see here, it's

Page 35

very clearly listed there are different things that identify a flow, like source and destination addresses, source and destination ports, and protocol, and session tokens derived 10:18:22 from the zone and virtual router.

Now, there are certain things here, like the source and destination addresses and the port numbers, the source and destination ports and protocol, that come on every packet, but 10:18:32 there are other things mentioned here, like virtual routers. In one instance of deployment, there may be a virtual router, in another instance, they may be not be using virtual routers. So, again --

10:18:42 Q. It would depend on the nature of the traffic?

A. No. It does not depend on the nature of the traffic. That's incorrect.

Q. What does it depend on?

10:18:48 A. It depends on how the -- the -- the admin has configured that particular box.

Q. Okay.

A. It does not depend on the nature of the traffic.

10:18:54 Q. Okay. So the admin has to configure the

Page 98

subsequent packets -- for that packet, as well as the subsequent packets of the flow to get the same treatment --

BY MR. HOSIE:

11:36:08 Q. So you don't have to --

A. -- as defined.

Q. -- go through -- yes. Thank you. So you don't have to go through this lookup process packet by packet by packet?

11:36:14 MR. MCPHIE: Objection. Asked and answered. Vague and ambiguous.

THE WITNESS: The flow state is stored in memory so that the policy lookup need not happen on a packet-by-packet basis for a given flow. 11:36:26

BY MR. HOSIE:

Q. Thank you. And that's the efficient way of doing it; right? In a flow-based model?

MR. MCPHIE: Objection. Vague and ambiguous. 11:36:34

THE WITNESS: That's one way of doing it.

BY MR. HOSIE:

Q. And that's how Juniper does it?

11:36:38 MR. MCPHIE: Objection. Vague and

Page 99

ambiguous.

THE WITNESS: In the J-series routers that we -- that we are currently talking about --

11:36:44 BY MR. HOSIE:

Q. Yeah.

A. -- that's how that is done.

Q. Okay. And the stateful firewall functionality, that's also part of the JUNOS operating system for the J-series routers? 11:36:48

A. So JUNOS operating systems has various modules. One of the modules is the stateful firewall processing.

Q. What's that module called?

11:37:12 A. The module has a name, Flow D. Stands for Flow Daemon.

Q. Daemon spelled D-a-e-m-o-n?

A. O-n. That's correct.

Q. Okay. And that's the name -- and so if I were to use your internal search tool and look for Flow Daemon, I would find that at the security firewall module? 11:37:26

A. That's correct.

Q. Okay. What other modules does JUNOS have? 11:37:38

Page 100

A. JUNOS has many different modules. It's very difficult for me to recall each and every module; right? We sell routers at the subscriber edge, you know, broadband subscriber 11:37:50 edge. So there's a Daemon that does authentication to the service provider's subscriber management system. I mean, I can go on and I --

Q. Let me ask a -- fair point. Let me ask 11:38:00 it a little more pointedly.

Is there a module that deals with class of service issues?

A. There is a Daemon that -- that is associated with the class of service, quality of 11:38:14 service, yes.

Q. Okay. So, you know, quality of service is a long-standing term; right?

A. Yes. Quality of service has been in the market since IP was invented.

Q. Of course.

A. Because there is the type of service that's built into the header.

Q. And -- yes. And I've seen a number of Juniper documents that don't use the generic 11:38:34 quality of service, or QOS, but, instead, call

Page 122

A. So it can go all the way up to Layer 7. Deep packet means you can look for attacks all the way from Layer 2 to Layer 7.

Q. That's right. Because you may have an
13:10:20 attack that is buried inside the payload itself, inside the content?

A. There could be attacks that are -- that are buried, and the protocol headers that are attacks that could be buried in -- in payload
13:10:32 content and, yeah, that's right.

Q. Okay. And IDP gives you a mechanism for trying to detect such payload-level attacks?

MR. MCPHIE: Objection. Vague and ambiguous.

13:10:44 THE WITNESS: I think that's a very broad statement. Payload -- I would like to constrain the -- the definition as data that is about the TCP layer. It could be payload; it could be headers. I would probably say, again,
13:11:06 payload with respect to what?

BY MR. HOSIE:

Q. Right. It might be payload, or it might not be? It might be Level 5, for instance?

A. Yeah. Right.

13:11:12 Q. Okay. Fair enough. If you could turn

Page 123

to page -- it ends with the digits 97 of the Viking Design Specification, there's a section 3.3 called "Packet Flow."

A. 3.3. Yes.

13:11:34 Q. Are you there, sir?

A. Yes, I am.

Q. Okay. So the SRX uses stateful flow processing?

A. We have different models of SRX
13:11:48 products. Some of the products have packet-based processing. Some of them have, in addition to packet-based processing, flow-based processing as well.

Q. Okay. Which modules have flow-based
13:12:00 processing for the SRX?

A. So if you broadly look at the SRX portfolio, you can divide that into two groups. There is a group which have a three-digit model number, those SRX models provide both
13:12:18 packet-based as well as flow-based functionality, and there is a group of SRX products, which are full -- full digit in model numbers, and those set of devices provide just the flow-based processing.

13:12:34 Q. Just flow-based alone?

Page 124

A. Just flow-based alone.

Q. So SRX three digits is both. SRX four digits is flow-based alone?

MR. MCPHIE: Objection. Compound.
13:12:42 Asked and answered.

THE WITNESS: SRX three-digit is packet and flow-based, and SRX four-digit is flow-based processing.

BY MR. HOSIE:

13:12:50 Q. Okay. So section 3.3 talks about packet flow. Is this for the flow-based aspects of those modules?

A. Section 3.3 is for the flow-based aspects of the system.

13:13:02 Q. Okay. So in terms of the basic flow-based architecture, does this differ from what we described this morning in the J-series routers?

A. This differs a little bit because of the
13:13:14 architecture from the one that we discussed in the morning.

Q. How does it differ, if you can tell me?

A. The J-series router is a single module from a physical hardware module, single-module
13:13:30 system, whereas the architecture that is

Page 201

I, BRENDA L. MARSHALL, Certified Shorthand Reporter, License No. 6939, do hereby certify:

That, prior to being examined, the witness named in the foregoing deposition, to wit, KRISHNA NARAYANASWAMY, was by me duly sworn to testify the truth, the whole truth and nothing but the truth:

That said transcript was taken down by me in shorthand at the time and place therein named and thereafter reduced to computerized transcription under my direction.

I further certify that I am not interested in the event of the action.

WITNESS this 30th day of September, 2011.

____

BRENDA L. MARSHALL

EXHIBIT 10

Page 1

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA

```
IMPLICIT NETWORKS, INC.,         )
                                 )
             Plaintiff,          )
                                 )
vs.                              )  No. C 10-4234 SI
                                 )
JUNIPER NETWORKS, INC.,          )
                                 )
             Defendant.          )
_________________________________)
```

DEPOSITION OF: PETER ALEXANDER, Ph.D.
TAKEN ON: October 16, 2012

13235

BEVERLY L. NEWMAN
CSR No. 2872

Page 187

BY MR. HOSIE: 02:54:12

Q What are the policy rules called in the SRX 02:54:13
context? 02:54:17

MR. McPHIE: Objection. Vague and ambiguous. 02:54:18

THE WITNESS: They could be called policy 02:54:21
rules. I'd have to check. 02:54:23

BY MR. HOSIE: 02:54:24

Q Do the SRX policy -- 02:54:24

MR. McPHIE: Were you finished with your 02:54:27
answer? 02:54:27

THE WITNESS: Well, I can take the time to 02:54:28
check, or you can ask another question if you want to. 02:54:29

BY MR. HOSIE: 02:54:31

Q I'll ask another question. 02:54:32

Sir, do the SRX policy rules function any 02:54:33
differently than the service sets for the multi-services 02:54:36
PIC? 02:54:41

MR. McPHIE: Objection. Compound. Vague and 02:54:42
ambiguous. Calls for speculation. 02:54:43

THE WITNESS: Yes. I mean, if you go and look 02:54:48
at the iRunway document that was provided recently -- 02:54:50

BY MR. HOSIE: 02:54:54

Q IRunway? 02:54:55

MR. McPHIE: The document from Povel you 02:54:57
produced a few days ago. 02:54:58

Page 188

BY MR. HOSIE: 02:55:00
Q What makes you think that's from iRunway? 02:55:00
A Oh, I'm sorry. I -- 02:55:02
MR. McPHIE: And the mistake would be ours. I 02:55:03
thought Povel or Mr. -- is it Treshkanov? I thought he 02:55:06
was from iRunway. 02:55:10
MR. HOSIE: He's not. It's not an iRunway 02:55:12
report. 02:55:15
MR. McPHIE: Oh, what is it? 02:55:15
BY MR. HOSIE: 02:55:15
Q So when you're referring to the iRunway, you 02:55:15
are referring to the Povel -- Provel work product; 02:55:18
correct? 02:55:19
A All right. I stand corrected. I received 02:55:21
incorrect information. 02:55:34
But the Provel report, which I believe 02:55:34
Dr. Nettles read and used, that has a clear 02:55:34
understanding of the way service sets are used, and 02:55:34
that's not the understanding, or the understanding you 02:55:36
get when you read the SRX product line documentation. 02:55:39
You are aware that there are no service sets, 02:55:48
and I don't believe they are discussed in that book that 02:55:51
you have. I'm not sure. But I don't recall seeing 02:55:55
them. And so service sets are defined on a basis and 02:55:59
provide a way of diverting traffic into a multi-services 02:56:09

Page 189

PIC element. 02:56:13

Q As a functional matter, do the policy rules in the SRX products work any differently than the service set rules in the multi-services PIC product? 02:56:15 02:56:18 02:56:20

MR. McPHIE: Objection. Compound. Vague and ambiguous. Calls for speculation. 02:56:23 02:56:25

THE WITNESS: All right. So you used the term "any different." Certainly they are very different because if you took service sets as they are defined in the multi-services product lines, there's no way you could run them in the other products, the accused products. 02:56:29 02:56:31 02:56:33 02:56:37 02:56:40 02:56:45

BY MR. HOSIE: 02:56:45

Q I'm asking about function, sir. And let me ask it again. 02:56:45 02:56:48

Do the policy rules in the SRX functionally work differently than the service sets and the PIC? 02:56:48 02:56:54

MR. McPHIE: Same objections. Asked and answered. Vague and ambiguous. Compound. Calls for speculation. 02:56:57 02:56:58 02:57:03

THE WITNESS: I don't see them as being directly related. They may be rules, but, you know, there's no discussion in any of the documentation or Dr. Nettles hasn't provided my discussion of how service sets might be used for SRX products. So I don't see 02:57:03 02:57:05 02:57:10 02:57:12 02:57:18

Page 190

that you can characterize them as not being different. 02:57:20
They are distinctly different. 02:57:26
BY MR. HOSIE: 02:57:27
Q Do you recall the question I asked you? 02:57:28
A No. You'll have to repeat it. 02:57:29
Q Sir, in terms of function, do the policy rules 02:57:31
in the SRX products work differently than the service 02:57:37
sets in the multi-services PIC? 02:57:41
MR. McPHIE: Objection. Asked and answered. 02:57:45
Compound. Vague and ambiguous. Calls for speculation. 02:57:46
THE WITNESS: You'll have to give me an example 02:57:54
of rules that you're talking about for all the SRX 02:57:56
product and the service set, and maybe I could answer it 02:57:58
by comparing them. 02:58:02
BY MR. HOSIE: 02:58:03
Q Did you do that work in the course of coming to 02:58:03
your conclusions in this case? 02:58:06
MR. McPHIE: Objection. Vague and ambiguous. 02:58:08
THE WITNESS: What I did is I tried to find 02:58:09
where SRX -- I'm sorry -- where service sets were used 02:58:11
in the SRX documentation. 02:58:16
If you let me borrow your book -- I think I had 02:58:18
trouble, and I don't think I actually found any 02:58:22
discussion in that book, so -- 02:58:25
///

Page 246

REPORTER'S CERTIFICATE

I, BEVERLY L. NEWMAN, CSR No. 2872, certify:

That the foregoing deposition of PETER ALEXANDER, Ph.D. was taken before me at the time and place therein set forth, at which time the witness was put under oath by me;

That the testimony of the witness and all objections made at the time of the deposition were recorded stenographically by me and were thereafter reduced to a computerized transcript under my direction;

That the foregoing transcript is a true record of the testimony of the witness and of all objections and colloquy made at the time of the deposition.

I further certify that I am neither counsel for nor related to any party to said action nor in anywise interested in the outcome thereof.

IN WITNESS WHEREOF, I have subscribed my name this 19th day of October, 2012.

____

BEVERLY L. NEWMAN, CSR No. 2872